

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»

институт

Прикладная информатика, математика и естественнонаучные дисциплины
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ Е.Н. Скуратенко

подпись инициалы, фамилия

« _____ » июня _____ 2016г.

БАКАЛАВРСКАЯ РАБОТА

_____ 09.03.03 «Прикладная информатика» _____

код – наименование направления

_____ Разработка калькулятора для потребительского кредита _____
_____ в Абаканском отделении №8602 Сбербанка России _____

тема

Руководитель

подпись, дата

ст. преподаватель
должность, ученая степень

_____ В.И. Кокова _____

инициалы, фамилия

Выпускник

подпись, дата

_____ А.А. Крылова _____

инициалы, фамилия

Нормоконтролер

подпись, дата

_____ В.И. Кокова _____

инициалы, фамилия

Абакан 2016

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
"Сибирский федеральный университет"

институт

Прикладная информатика, математика и естественнонаучные дисциплины
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ Е.Н. Скуратенко

подпись инициалы, фамилия

« ____ » _____ 2016 г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту Крыловой Ангелине Алдын-ооловне

фамилия, имя, отчество

Группа 52-1 (ХБ 12-06) Направление (специальность) 09.03.03

номер

код

«Прикладная информатика»

Наименование

Тема выпускной квалификационной работы Разработка калькулятора для потребительского кредита в Абаканском отделении №8602 Сбербанка России

Утверждена приказом по институту № 170 от 01.03.2016

Руководитель ВКР В.И. Кокова, ст. преподаватель, ХТИ – филиал СФУ

инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР Документы Абаканского отделения №8602 Сбербанка России

Перечень разделов ВКР: _____

1. Моделирование деятельности организации ПАО «Сбербанк России».

2. Разработка кредитного калькулятора для Абаканского отделения №8602 Сбербанка России.

3. Расчет показателей экономической эффективности от внедрения потребительского калькулятора в Абаканском отделении №8602 Сбербанка России.

Перечень графического материала – _____

Руководитель ВКР _____

подпись

В.И. Кокова

инициалы и фамилия

Задание принял к исполнению _____

подпись,

А.А. Крылова

инициалы и фамилия студента

«01 » марта 2016 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка калькулятора для потребительского кредита в Абаканском отделении №8602 Сбербанка России» содержит 83 страниц текстового документа, 15 таблиц, 24 рисунков, 27 формул, 6 приложений, 19 использованных источников.

СБЕРБАНК, КРЕДИТОВАНИЕ, ПОТРЕБИТЕЛЬСКИЙ КРЕДИТ, UML, МОБИЛЬНЫЙ КАЛЬКУЛЯТОР, ПРИЛОЖЕНИЕ, ANDROIDSTUDIO, DELPHI, HTML, JAVASCRIPT, CSS, JQUERY, JAVA, XML.

Объект исследования – потребительский кредит в Абаканском отделении №8602 Сбербанка России.

Цель работы: улучшение качества обслуживания клиентов Абаканского отделения №8602 Сбербанка России за счет разработки потребительского калькулятора.

Для достижения цели необходимо решить следующие задачи: проанализировать деятельность кредитного отдела, построить диаграммы вариантов использования информационной системы отдела кредитования, разработать калькулятор для потребительского кредита Абаканскому отделению №8602 Сбербанка России, рассчитать экономический эффект от внедрения разработки калькулятора и ее реализации.

В результате анализа деятельности кредитного отдела было выявлено отсутствие потребительского кредитного калькулятора для мобильного телефона, которым можно пользоваться в любом месте и без доступа в интернет.

В итоге были разработаны мобильный и стационарный калькуляторы для потребительского кредита, позволяющие улучшить качество обслуживания клиентов.

ABSTRACT

Graduation thesis on the theme of «Calculator development for a consumer credit at Sberbank (Abakan office) in favour of branch № 8602» contains 83 pages of a text document, 15 tables, 27 formulas, 6 appendices, the list of 19 references.

SBERBANK, CREDITING, CONSUMER CREDIT, UML, MOBILE CALCULATOR, APPLICATION, ANDROID STUDIO, DELPHI, HTML, JAVASCRIPT, CSS, JQUERY, JAVA, XML.

The object is a consumer credit at Sberbank (Abakan office) in favour of branch № 8602.

The aim is to improve customer service quality at Sberbank (Abakan office) in favour of branch № 8602 using consumer calculator development.

To achieve this aim, it needs to solve the following tasks: to analyze activities of the credit department; to make diagrams of information system use cases for the credit department; to develop a calculator for a consumer credit at Sberbank (Abakan office) in favour of branch № 8602; to estimate economic efficiency from consumer calculator implementation.

Activities analysis of the credit department shows no-absence of a consumer calculator for mobile phone which can be used anywhere without Internet access.

As a result, it is developed a mobile and stationary calculators for consumer credits to improve consumer service quality.

Normkontroller of foreign language

signature, date

Elokhova G.V.
(surname, name, patronymic)

СОДЕРЖАНИЕ

Введение.....	8
1 Моделирование деятельности организации ПАО «Сбербанк России».....	10
1.1 Деятельность организации ПАО «Сбербанк России».....	10
1.2 Структурная схема организации ПАО «Сбербанк России»	14
1.3 Модель диаграммы вариантов использования информационной системы отдела кредитования в ПАО «Сбербанк России».....	17
1.4 Теоретические основы организации кредитования ПАО «Сбербанк России».....	30
1.4.1 Функции, принципы и формы кредита	30
1.5 Постановка цели работы. Формулировка задач, подлежащих решению	33
1.6 Анализ литературы и других источников информации по разработке кредитного калькулятора.....	34
1.7 Обоснование выбора средств для разработки приложений	37
1.8 Выводы по разделу «Моделирование деятельности организации ПАО «Сбербанк России»»	40
2 Разработка кредитного калькулятора для Абаканского отделения №8602 Сбербанка России.....	42
2.1 Функциональная архитектура «Потребительский калькулятор»	42
2.2 Файловая структура мобильного и стационарного калькулятора	43
2.3 Описание программных модулей	47
2.4 Описание пользовательского интерфейса «Потребительский калькулятор».....	49

2.5 Вывод по разделу «Разработка кредитного калькулятора для Абаканского отделения №8602 Сбербанка России»	59
3 Расчет показателей экономической эффективности от внедрения потребительского калькулятора в Абаканском отделении №8602 Сбербанка России	60
3.1 Расчет трудозатрат на разработку программного продукта «Потребительский калькулятор».....	60
3.2 Экономическая оценка разработки программного продукта «Потребительский калькулятор».....	65
3.3 Расчет эффективности внедрения разработанных программ	75
3.4 Выводы по экономическому разделу «Расчет показателей экономической эффективности от внедрения потребительского калькулятора в Абаканском отделении №8602 Сбербанка России»	79
Заключение	80
Список сокращений	82
Список использованных источников	83
Приложения А – Е.....	85

ВВЕДЕНИЕ

В выпускной квалификационной работе рассматривается объект исследования – отдел кредитования Абаканского отделения №8602 ПАО «Сбербанк России».

Цель работы: улучшение качества обслуживания клиентов Абаканского отделения №8602 Сбербанка России за счет разработки потребительского калькулятора.

Для достижения цели необходимо решить следующие задачи:

- Проанализировать деятельность кредитного отдела.
- Построить диаграмму вариантов использования информационной системы отдела кредитования в Абаканском отделении №8602 Сбербанка России.
- Разработать калькулятор для потребительского кредита Абаканскому отделению №8602 «Сбербанка России».
- Рассчитать экономический эффект от внедрения разработки калькулятора и ее реализации.

В результате анализа деятельности кредитного отдела было выявлено отсутствие потребительского кредитного калькулятора для мобильного телефона, которым можно пользоваться в любом месте.

В итоге будут разработаны мобильный и стационарный калькуляторы для потребительского кредита, позволяющие улучшить качество обслуживания клиентов.

Разработка потребительского мобильного калькулятора для расчета кредита является актуальной, так как им можно пользоваться в любом месте. Проанализированы существующие кредитные калькуляторы и выяснено, что у других калькуляторов включена только одна функция (например: расчет ежемесячного платежа или же расчет пени), в выпускной квалификационной работе будет разработан такой потребительский

калькулятор, в который включены необходимые функции для расчета кредита, также для улучшения использования будут подсказки.

В первом разделе рассматривается деятельность организации ПАО «Сбербанк России», построена диаграмма вариантов использования информационной системы отдела кредитования, где представлена информация, как происходит кредитование физическим и юридическим лицам (индивидуальных клиентов и организаций). Выявляется проблема с расчетом сумм кредитов для потребителей. Проводится анализ используемых программных продуктов в отделе кредитования. Дано обоснование решения разработки собственной разработки кредитного калькулятора для потребителей.

Второй раздел представляет описание и реализацию решения выявленной проблемы, отсутствие потребительского кредитного калькулятора для мобильного телефона с необходимыми для потребителя функциями. Описана разработка калькулятора потребительского кредита для Абаканского отделения №8602 ПАО «Сбербанк России».

В основе третьего раздела выпускной квалификационной работы лежит анализ единовременных затрат для разработки кредитного калькулятора для потребителей и обоснование экономического эффекта от внедрения кредитного калькулятора в организацию ПАО «Сбербанк России».

1 Моделирование деятельности организации ПАО «Сбербанк России»

1.1 Деятельность организации ПАО «Сбербанк России»

Полное наименование организации – публичное акционерное общество «Сбербанк России».

Юридический адрес: Республика Хакасия, г. Абакан, ул. Пушкина, 165.

Почтовый адрес: 655017, Республика Хакасия, г. Абакан, ул. Пушкина, 165.

Вид собственности: Смешанная российская собственность с долей федеральной собственности.

Отрасль, выпускаемая продукция (работа, услуги): предоставление банковских услуг частным и корпоративным клиентам: кредитование, депозиты, валютные операции, денежные переводы и т.д.

Краткое описание организации. «Сбербанк России» — российский коммерческий банк, международная финансовая группа, один из крупнейших банков России и Европы. Действует в интересах вкладчиков, клиентов и акционеров, банк стремится эффективно инвестировать привлеченные средства частных и корпоративных клиентов в реальный сектор экономики, кредитует граждан, способствует устойчивому функционированию российской банковской системы и сбережению вкладов населения. Основные направления деятельности банка: максимальная ориентация на клиента и превращение Сбербанка в «сервисную» компанию по обслуживанию индивидуальных и корпоративных клиентов; технологическое обновление банка и «индустриализация» систем и процессов; существенное повышение операционной эффективности банка на основе самых современных технологий, методов управления,

оптимизации и рационализации деятельности по всем направлениям за счет внедрения производственной системы Сбербанка, разработанной на базе технологий Lean; развитие операций на международных рынках, прежде всего в странах СНГ.

История развития ПАО «Сбербанк России» началась 12 ноября 1841 года. Российским императором Николаем I был подписан указ об учреждении в России сберегательных касс «для доставления через то средств к сбережению верным и выгодным способом». Эта дата стала считаться днем рождения Сбербанка.

Сбербанк является историческим преемником основанных указом императора Николая I Сберегательных касс, которые поначалу были лишь двумя маленькими учреждениями с 20 сотрудниками в Санкт-Петербурге и Москве. Затем они разрослись в сеть сберегательных касс, работавших по всей стране и даже в трудные времена помогавших сохранить устойчивость российской экономики. Позже, в советскую эпоху, они были преобразованы в систему Государственных трудовых сберегательных касс. А в новейшее время превратились в современный универсальный банк, крупную международную группу, чей бренд известен более чем в двадцати странах мира [12].

Ниже в таблице 1 представлены этапы развития ПАО «Сбербанк России».

Таблица 1—Этапы развития ПАО «Сбербанк» России

Этапы	Дата	Факты
1	1841 —1895	Основание и развитие банковского дела в России. Первый клиент первого банка страны. Рост до 500 клиентов в день

Продолжение таблицы 1

Этапы	Дата	Факты
2	1895 — 1917	«Золотой век» первого банка России и развитие финансовой грамотности населения. — 3875 сберегательных касс — 2 000 000 выданных сберкнижек
3	1917 — 1941	Первая революция и глобальные перемены в политике Сбербанка. Сберкассы стали осуществлять денежные переводы, выпускать собственные заемные сертификаты, проводили операции с процентными и ценными бумагами.
4	1941 — 1953	Сбербанк в годы Великой Отечественной войны и послевоенное время, участие в проектах государственного и общемирового значения. Учрежден новый Устав государственных сберкасс.
5	1953 — 1991	Развитие и преобразование Сбербанка во времена «оттепели», «застоя» и «перестройки». Количество сберегательных касс увеличилось вдвое: с 40 тысяч до 79 тысяч. Количество клиентов банка возросло в 12 раз.
6	1991 — 2008	Глобальные перемены «Сбербанка России»: жизнь по новым экономическим законам. Начали функционировать первые банкоматы. Создан Негосударственный пенсионный фонд Сбербанка. Начала работать услуга «Сбербанк ОнЛайн».
7	2009	Деятельность и меры Сбербанка России в тяжелой финансовой ситуации: кризис преодолен. Начала работать система «Сбербанк бизнес ОнЛайн». Он вошел в топ-20 крупнейших банков по рыночной капитализации. Стал генеральным партнером Олимпийских игр «Сочи-2014».

Окончание таблицы 1

Этапы	Дата	Факты
8	2010	<p>Новый этап в истории Сбербанка России: внедрение инновационных решений, новые программы и прогрессивные технологии. Новое будущее страны.</p> <p>Сбербанк отменил все комиссии за рассмотрение и выдачу кредитов. Дважды были снижены процентные ставки кредитования.</p>

Новое полное фирменное наименование Банка на русском языке: Публичное акционерное общество «Сбербанк России». Новое сокращенное фирменное наименование Банка на русском языке: ПАО «Сбербанк».

Публичное акционерное общество «Сбербанк России» (ПАО «Сбербанк России») — это современная универсальная организация, входящая в число крупнейших российских коммерческих банков России и стран СНГ [17].

Сбербанк России имеет огромную филиальную сеть: 17 территориальных банков и более 18 400 подразделений. Он оказывает услуги во всех 83 субъектах Российской Федерации. Недавно были введены удаленные каналы обслуживания. Происходит развитие приложений «Сбербанк Онлайн» и «Мобильный банк» с широкой клиентской базой.

Также в последнее время «Сбербанк» значительно расширил зону международного присутствия. Ранее она ограничивалась странами СНГ, однако в последнее время зона распространения существенно возросла. Появились представительства в Центральной и Восточной Европе (Sberbank EuropeAG) и в Турции (DenizBank). Приобретение DenizBank оказалось самым значительным за всю 170-летнюю историю банка. Помимо ранее перечисленных стран, «Сбербанк» имеет еще представительства в Китае, Индии и Германии, управляет Sberbank Switzerland AG.

Сбербанк России зарегистрирован 20 июня 1991 год в Центральном Банке Российской Федерации.

Сбербанк России создан в форме акционерного общества открытого типа в соответствии с Законом РСФСР «О банках и банковской деятельности в РСФСР» в 1991 году. Учредителем и основным акционером Сбербанка России является Центральный банк Российской Федерации. Акционерами Банка являются более 200 тысяч юридических и физических лиц.

1.2 Структурная схема организации ПАО «Сбербанк России»

На организационную структуру банка влияет прежде всего характер банковской деятельности и регламентируется структура Уставом банка.

Высшим органом управления банком является общее собрание акционеров банка, которому подотчетны избираемые им ревизионная комиссия и совет директоров, осуществляющий общее руководство деятельностью банка в периоды между общими собраниями.

Исполнительным органом управления выступает правление банка, возглавляемое его председателем, в непосредственном ведении которого могут находиться центральные отделы (планирования, аналитический, юридический, маркетинговый и другие), а также вспомогательные службы (канцелярия, службы безопасности и другие).

Совещательным органом при правлении банка является кредитный комитет.

В ведении соответствующих заместителей председателя правления обычно находятся коммерческие отделы, деятельность которых связана с получением доходов (кредитный, валютный, ценных бумаг, расчетно-кассового обслуживания и т.д.).

По мере роста банковских операций, расширения операций на финансовом рынке и освоения новых рынков, внедрения новых банковских продуктов и технологий коммерческие банки совершенствуют свои организационные структуры с учетом изменившихся условий банковской деятельности.

Более подробно структура коммерческого банка представлена на рисунке 1.

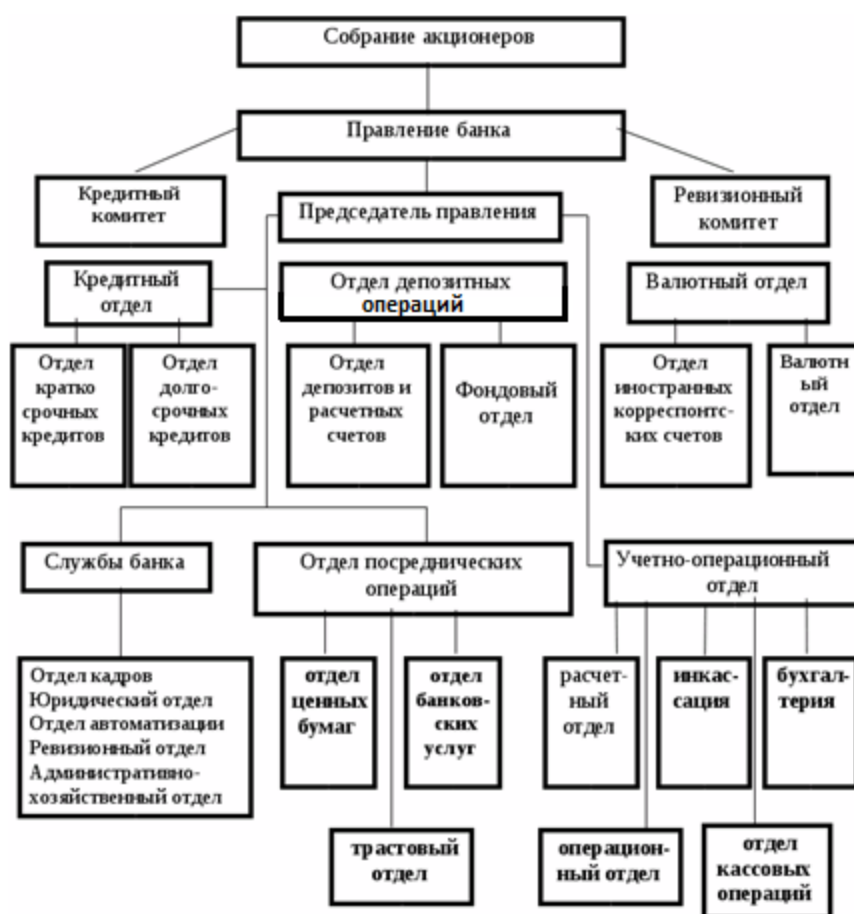


Рисунок 1 – Организационная структура ПАО «Сбербанк России»

Абаканское отделение № 8602 Сбербанка России имеет филиальную сеть, которая состоит из 6 Дополнительных офисов и 22 Операционных касс вне кассовых узлов, оказывающих как стандартный набор услуг, так и специализирующихся на развитии отдельных направлений бизнеса и на

индивидуальном обслуживании клиентов, расположены во всех районах и крупных населённых пунктах Республики Хакасия и предлагают клиентам полный комплекс услуг Сбербанк.

Отделение является структурным подразделением Восточно-Сибирского банка, структура управления – централизованная, деление на отделы осуществляется по функциональному признаку, с элементами дивизиональной структуры.

Управление в банке и отделение в частности, осуществляется централизованно, что имеет как недостатки, так и достоинства.

Высший орган управления – общее собрание акционеров, которое утверждает Устав, избирает Наблюдательный совет, принимает другие решения. Наблюдательный Совет осуществляет руководство деятельностью Сбербанка, определяет приоритеты его развития и формирует Правление, которому делегированы полномочия по текущему управлению.

Система управления; администрация; управляющий; заместители управляющего (2 штатные единицы).

В состав отделения входят структурные подразделения, каждое из которых по своим направлениям курируют деятельность ВСП (внутренние структурные подразделения) – операционные кассы и дополнительные офисы.

Основные из них: отдел кредитования; сектор вкладов и расчётов населения; сектор расчётно-кассового обслуживания юридических лиц и бюджетов; сектор валютных, неторговых операций и ценных бумаг, сектор банковских карт (рисунок 2).

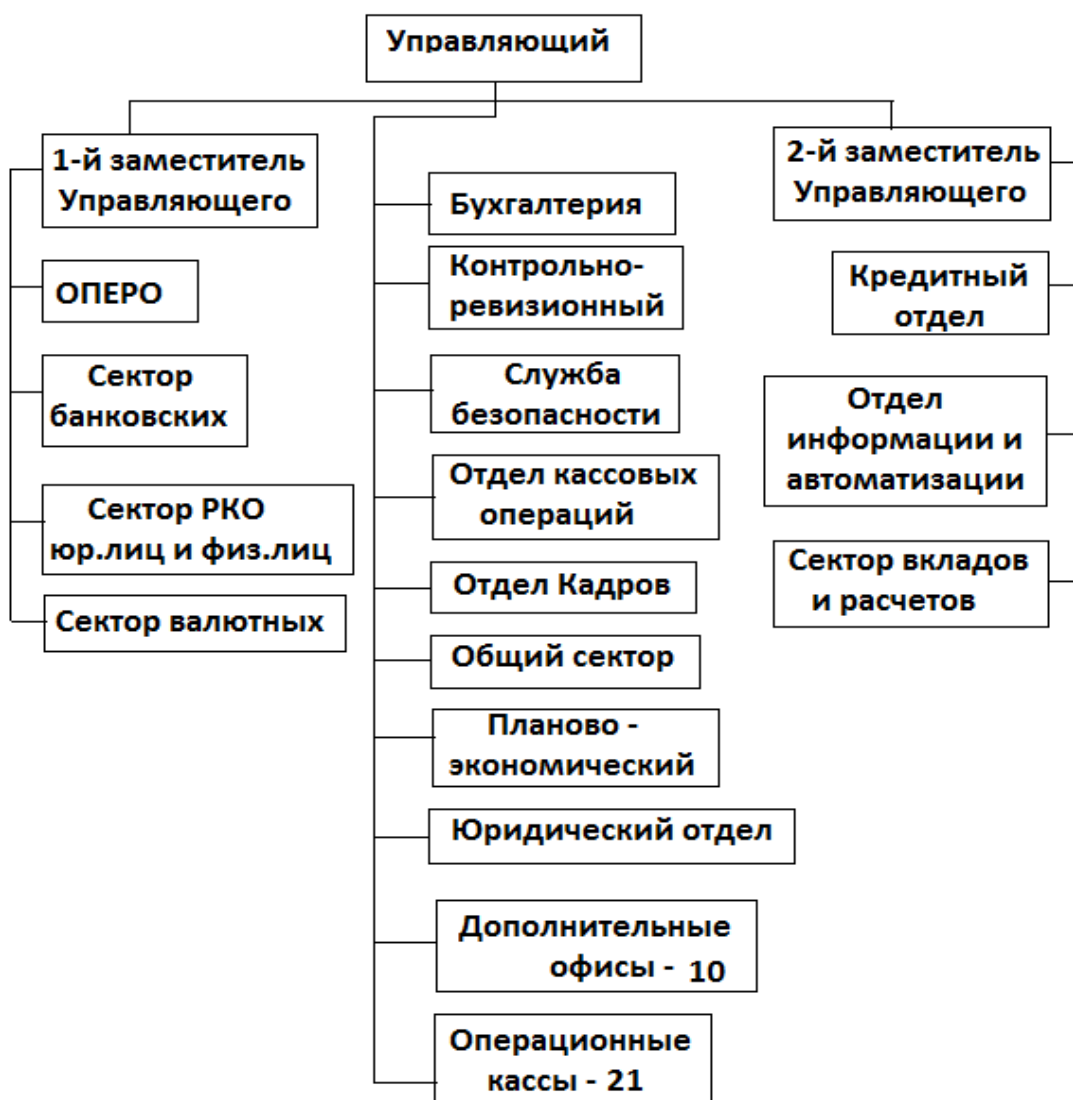


Рисунок 2 – Организационная структура Абаканского отделения №8602 ПАО «Сбербанк России»

1.3 Модель диаграммы вариантов использования информационной системы отдела кредитования в ПАО «Сбербанк России»

Сбербанк автоматизирует деятельность, связанную с кредитованием физических и юридических лиц (индивидуальных клиентов и организаций).

В настоящее время кандидат на получение кредита заполняет бумажную форму, прикладывает необходимые документы (финансовый

отчет, перспективную оценку финансового состояния и др.) и отправляет в банк. Референт по кредитованию анализирует запрос на предмет возможных ошибок и подтверждает его достоверность.

Затем референт запрашивает отчет о кредитных операциях клиента в отделе кредитования. Копия отчета просматривается банковским служащим, а референт проверяет финансовое положение и доход клиента. Служащий также обращается к существующей системе управления счетами клиентов, чтобы получить необходимую информацию о состоянии счета и предыдущим кредитам клиента.

Вся информация комплектуется в кредитный запрос и направляется для оценки инспектору по кредитам. Если запрос утверждается, инспектор определяет наилучшие условия кредитования и уведомляет об этом клиента. Если клиент принимает условия, то кредит оформляется.

На обработку запроса обычно уходит минимум две недели (как для индивидуальных клиентов, так и для организаций).

Цель автоматизации – сократить время обработки запроса до 2 часов для индивидуальных клиентов и 48 часов для организаций, сократить количество сотрудников, занятых в процессе обработки, и увеличить количество запросов, обрабатываемых в заданный период.

Действующие лица при кредитовании.

Предварительно составим список действующих лиц:

— Клиент – заполняет бумажную форму. Прикладывает необходимые документы и отправляет в банк. Принимает условие кредитования.

— Референт – анализирует запрос на предмет возможных ошибок и подтверждает его достоверность. Запрашивает отчет о кредитных операциях клиента в отделе кредитования. Проверяет финансовое положение и доход клиента.

— Банковский служащий – просматривает копии отчета о клиенте. Обращается к существующей системе управления счетами клиентов, чтобы получить необходимую информацию о состоянии счета и предыдущим кредитам клиента.

— Инспектор – определяет наилучшие условия кредитования и уведомляет об этом клиента.

Составление списка вариантов использования.

Вариант использования представляет собой последовательность действий (транзакций), выполняемых банковской системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Вариант использования описывает типичное взаимодействие между пользователем и системой. Каждый вариант использования – это потенциальное требование к системе, и пока оно не выявлено, невозможно запланировать его реализацию. Таким образом, список вариантов использования к банковской системе можно представить следующим образом:

– Заполнить форму на кредит. Действующее лицо «Клиент» вносит ФИО, паспортные данные и контакты на форму, предназначенную для кредита.

– Проанализировать запрос клиента (осуществляется референтом).

– Проверить финансовое положение клиента и его доход. Действующее лицо «Референт» анализирует запрос на предмет возможных ошибок и подтверждает его достоверность.

– Запросить отчет действующей системе о кредитных операциях клиента. Референт делает копию отчёта и передаёт её банковскому служащему.

– Запросить систему о состоянии счета клиента и его предыдущих операциях. Действующее лицо «Банковский служащий» запрашивает кредитную историю клиента и проверяет состояние счёта.

- Просмотреть копию отчета референта (осуществляется служащим банка).
- Получить скомпонованный кредитный запрос (осуществляется инспектором по кредитам).
- Уведомить клиента о решении банка (осуществляется инспектором по кредитам).
- Определить условия кредитования (осуществляется инспектором по кредитам).
- Утвердить кредит (осуществляется инспектором по кредитам).
- Принятие условий кредитования (осуществляется клиентом).
- Оформить кредит (осуществляется инспектором по кредитам).

Отообразим список вариантов использования с действующими лицами (рисунок 3).

На диаграмме вариантов использования показано взаимодействие между вариантами использования и действующими лицами. Данная диаграмма отражает взаимодействие между вариантами использования и действующими лицами банковской системы. Диаграмма показывает, какие действующие лица инициируют варианты использования. Также видно, когда действующее лицо получает информацию от варианта использования.

Описание варианта использования «Заполнить форму на кредит».

Данный вариант использования описывает заполнение пользователем формы на кредит и отправления ее в банк.

Основной поток событий:

1. Ввести данные о клиенте.
2. Определить есть ли ошибки.
3. Если ошибки есть, то выполняется альтернативный поток A1.
4. Если клиент уже есть в базе, то выполняется альтернативный поток событий E1.
5. Сохранить данные о клиенте.

6. Ввести данные о кредите.
7. Сохранить данные о кредите.

Альтернативный поток А1.

1. Информировать клиента, что данные введены с ошибками.
2. Очистить поля, в которые введены неверные данные.
3. Предложить ввести корректные данные.
4. Вариант использования завершен.

Альтернативный поток Е1.

1. Вывести данные о клиенте.
2. Вывести кредитную историю.
3. Проанализировать кредитную историю.
4. Если имеется задолженность, то выполняется альтернативный

поток В1.

Вариант использования «Оформить кредитный договор».

Данный вариант использования описывает оформление кредитного договора в соответствии с данными о клиенте.

Основной поток событий:

1. Открыть форму заявки на кредит.
2. Проверить данные о клиенте.
3. Проверить данные заявки о кредите.
4. Открыть форму «Формирование кредитного договора».
5. Заполнение дополнительных данных по кредитному договору (Сумма кредита, продолжительность гашения, процентная ставка, гарантии кредита, порядок гашения, дополнительные обязанности сторон).
6. Сформировать кредитный договор.
7. Просмотр кредитного договора.
8. Печать кредитного договора.

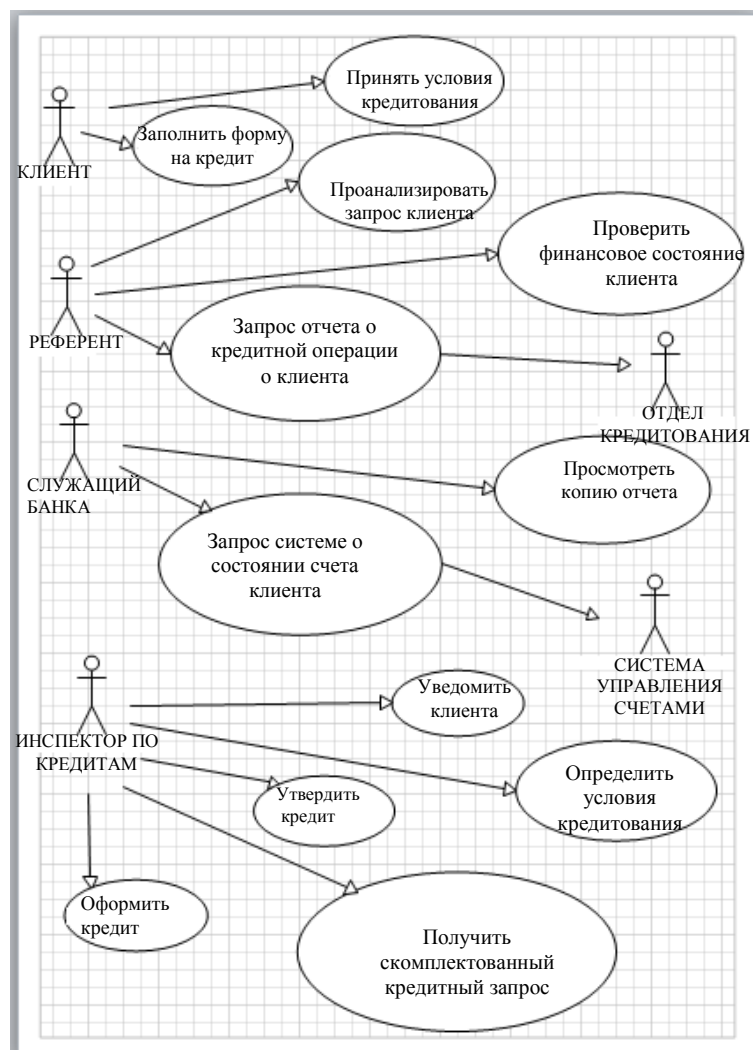


Рисунок 3 – Диаграмма вариантов использования банковской системы

При рассмотрении запроса клиента, референт отправляет запрос о кредитной операции отделу кредитования, Служащий банка в свою очередь просматривает копию отчета референта и делает запрос в систему управления со счетами, чтобы проверить состояние счета клиента. Инспектор по кредитам уведомляет клиента и определяет условия кредитования, клиент в свою очередь принимает условия и вид кредитования: аннуитентный или дифференцированный платеж [16].

Аннуитетный платеж – вариант ежемесячного платежа по кредиту, когда размер ежемесячного платежа остаётся постоянным на всём периоде кредитования.

Чаще применяется аннуитет: выгоднее для банка и удобнее для большинства клиентов.

Ежемесячная сумма аннуитетного платежа рассчитывается по формуле:

$$A = K * S, \quad (1)$$

где A – сумма платежа по кредиту;

K – коэффициент аннуитетного платежа;

S – величина займа.

Формула коэффициента аннуитета:

$$K = \frac{(i*(1+i)^n)}{((1+i)^n-1)}, \quad (2)$$

где i – процентная ставка за месяц, результат деления годовой ставки на 12;

n – срок кредита в месяцах.

Дифференцированный платеж – вариант ежемесячного платежа по кредиту, когда размер ежемесячного платежа по погашению кредита постепенно уменьшается к концу периода кредитования.

Дифференцированный способ оплаты предполагает, что:

— сумма основного долга распределена по периодам выплат равными долями;

— проценты по кредиту начисляются на остаток.

Формула расчета дифференцированного платежа:

$$ДП = \frac{ОСЗ}{(пп+ЩСЗ*ПС)}, \quad (3)$$

где $ДП$ – ежемесячный платеж по кредиту;

$ОСЗ$ – остаток займа;

ПП – число оставшихся до конца срока погашения периодов;

ПС – процентная ставка за месяц (годовую ставку делим на 12).

Клиент, выбрав удобный ежемесячный платеж, приносит все необходимые документы инспектору по кредитам, он получает скомплектованный кредитный запрос от клиента и оформляет кредит.

Вариант использования «Запрос системе о состоянии счета клиента».

Данный вариант использования описывает в каком состоянии находится счёт клиента.

Основной поток событий:

1. Войти в систему.
2. Открыть форму, предназначенную для управления счетами клиентов.
3. Внести ФИО клиента, паспортные данные и контакты.
4. Получить данные о кредите, кредитной истории.
5. Передать данные инспектору по кредитам.

Диаграммы последовательностей для варианта использования «Заполнить форму на кредит».

Все действующие лица показаны в верхней части диаграммы (Клиент). Объекты, требуемые системе для выполнения варианта использования «Форма на кредит» (граничный) и «Клиенты (сущность)». Каждый объект изображается в виде прямоугольника с указанием имени. Под каждым объектом рисуется вертикальная пунктирная линия. Ее называют линией жизни объекта. Далее для выполнения требуемых функций необходимы стрелки. Стрелки соответствуют сообщениям, передаваемым между действующим лицом «Клиент» и объектом «Форма на кредит» и между объектами «Форма на кредит» и «Клиенты». Сообщение нумеруются:

1. Открыть форму.
2. Заполнить форму.

3. Проверка данных.
4. Сохранение данных о клиенте.
5. Закрывать форму.
6. Открытие формы.

Первое сообщение показывается на самом высоком уровне, второе ниже и т.д. Линии, обозначающие посылку сообщения, проводятся от вертикали одного объекта к вертикали другого объекта. События помогают определить границы системы.

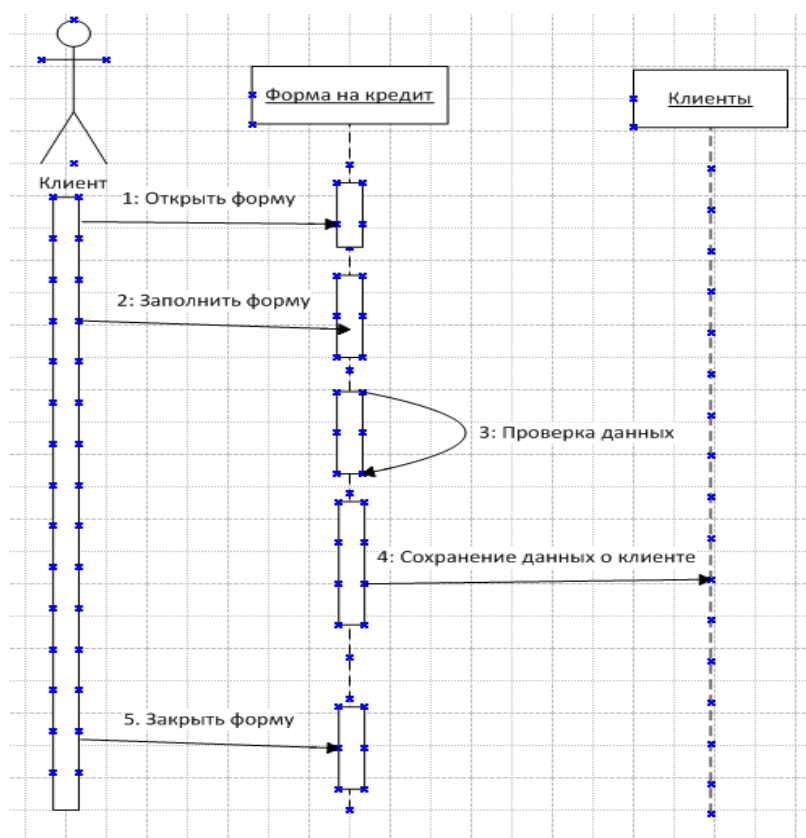


Рисунок 4 – Диаграмма последовательности для варианта использования «Заполнить форму на кредит»

Таким образом, построена диаграмма последовательности, изображенная на рисунке 4, отображает поток событий в рамках варианта использования «Заполнить форму на кредит».

Диаграммы последовательностей для варианта использования «Заполнить форму на кредит» – альтернативный поток A1

Для построения используется действующее лицо «Клиент», граничный объект «Форма на кредит» и объект-сущность «Ошибка».

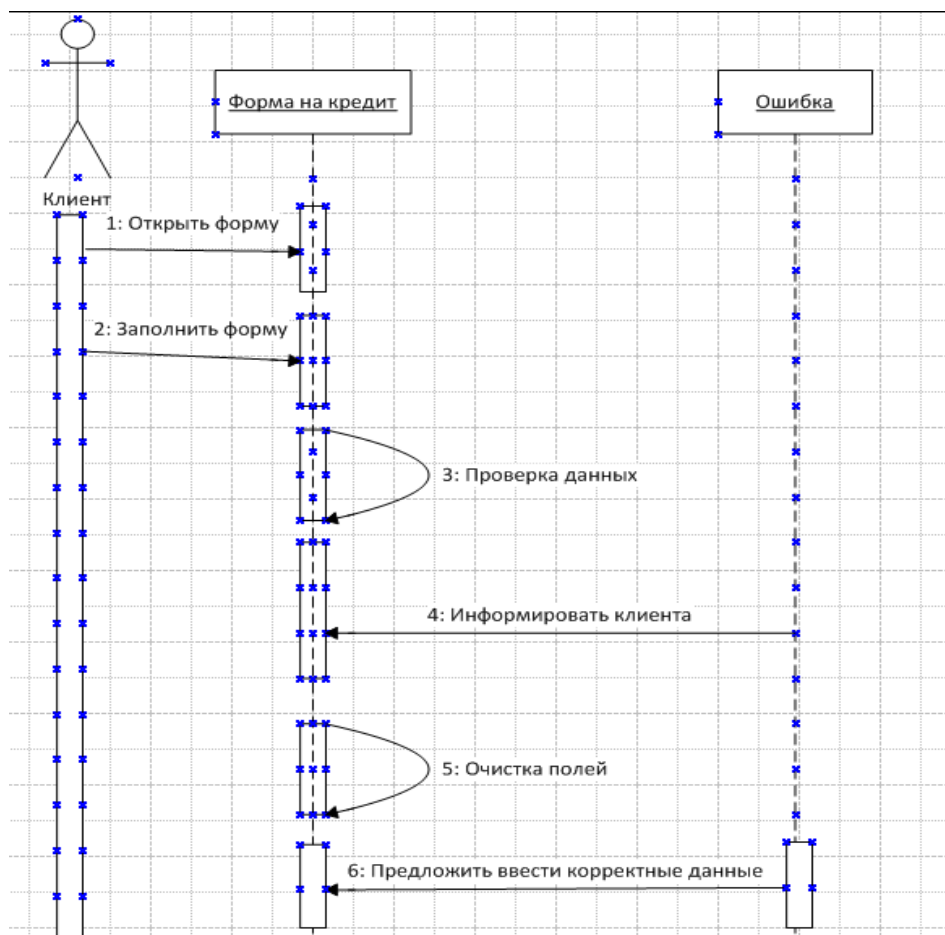


Рисунок 5 – Диаграмма последовательности для варианта использования «Заполнить форму на кредит» – альтернативный поток A1

Диаграмма последовательности для варианта использования «Оформить кредит».

Действующее лицо «Инспектор». Его задача определить наилучшие условия кредитования для клиента. Для этого ему потребуется три объекта «Копия отчёта референта» (граничный), «Договор на кредит»(граничный) и «Клиент»(сущность). Отметим, что линия жизни у объекта «Копия отчёта

референта» будет значительно меньше других объектов, так как он нужен только для просмотра. Сообщения, которые потребуются для выполнения функции:

1. Оценка запроса.
2. Просмотр копии.
3. Определение наилучших условий кредитования.
4. Уведомление клиента.
5. Принятие решения.
6. Сообщить о выбранном условии.
7. Оформление договора.

Построение диаграммы последовательности для варианта использования «Оформить кредит» на этом закончено (рисунокб).

Диаграммы классов.

Диаграмма классов (classdiagram) служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Класс (class) в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. В банковской системе определим четыре класса, таких как, «Клиент», «Референт», «Банковский служащий», «Инспектор», «Отдел кредитования» и «Система управления счетами клиентов». Атрибуты класса определяют связанные с классом данные. Тип атрибута String (Строка) означает, что отдельное значение имени представляет собой строку текста из одного или двух слов, например, фамилия имя отчество, дата рождения. Необходимые атрибуты и методы классов определяются на основе анализа диаграмм последовательностей «Оформить кредит» и «Заполнить форму на кредит», описания вариантов использования для банковской системы.

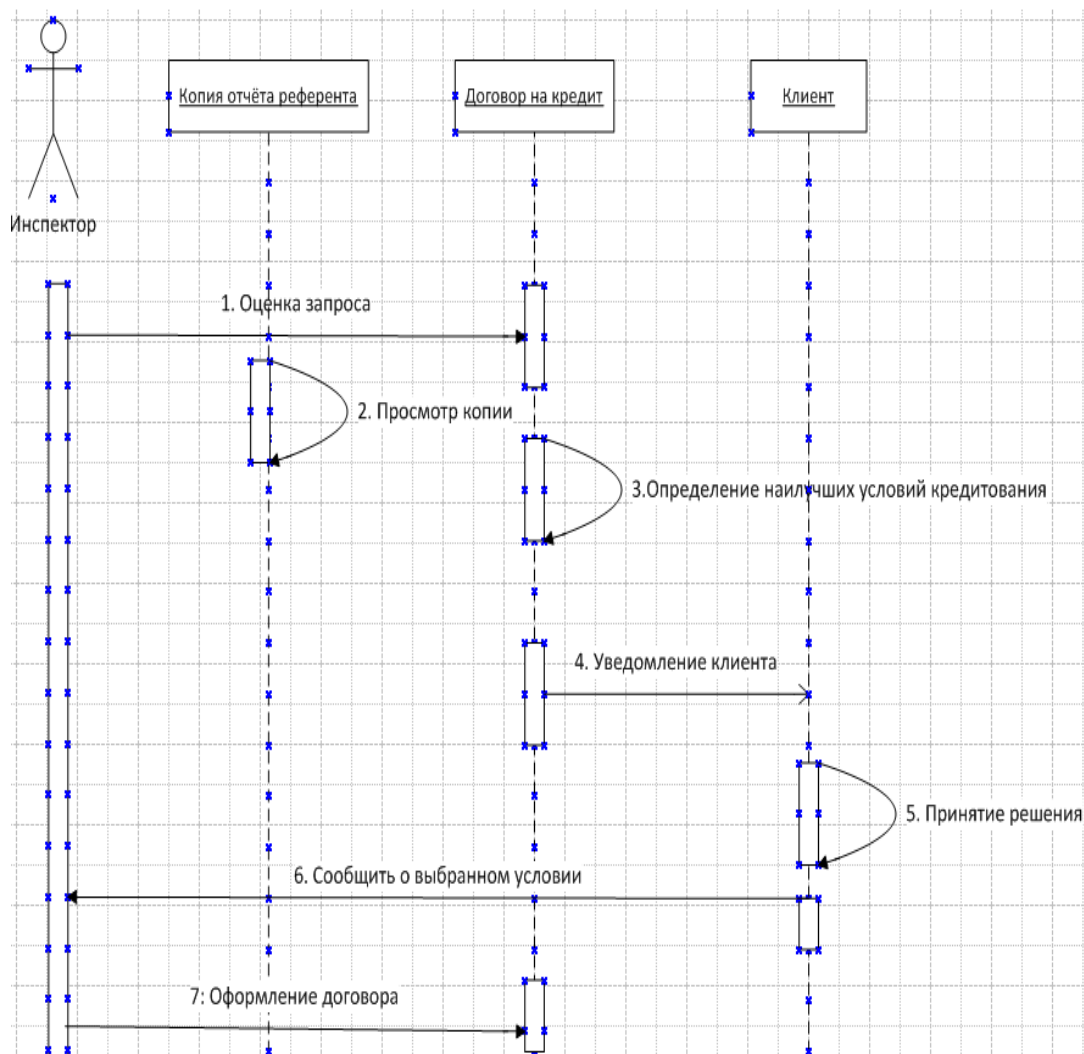


Рисунок 6 – Диаграммы последовательности для варианта использования «Оформить кредит»

Операции или методы класса записываются в третьей сверху секции прямоугольника. Операция (operation) представляет собой некоторый сервис, предоставляющий каждый экземпляр класса по определенному требованию. Совокупность операций характеризует функциональный аспект поведения класса.

Методы класса определяют функциональность класса. Существуют следующие методы (операции) классов:

– Операции доступа, в нашем случае символ «+» обозначает операцию с областью видимости типа общедоступный (public) квантор видимости.

- Об операции реализации, выполняют функции класса.
- Управляющие операции управляют созданием и уничтожением объектов. В эту категорию попадают конструкторы и деструкторы классов.

На рисунке 7 приведен фрагмент диаграммы классов для банковской системы. В данной диаграмме атрибуты и операции определены для четырёх классов.

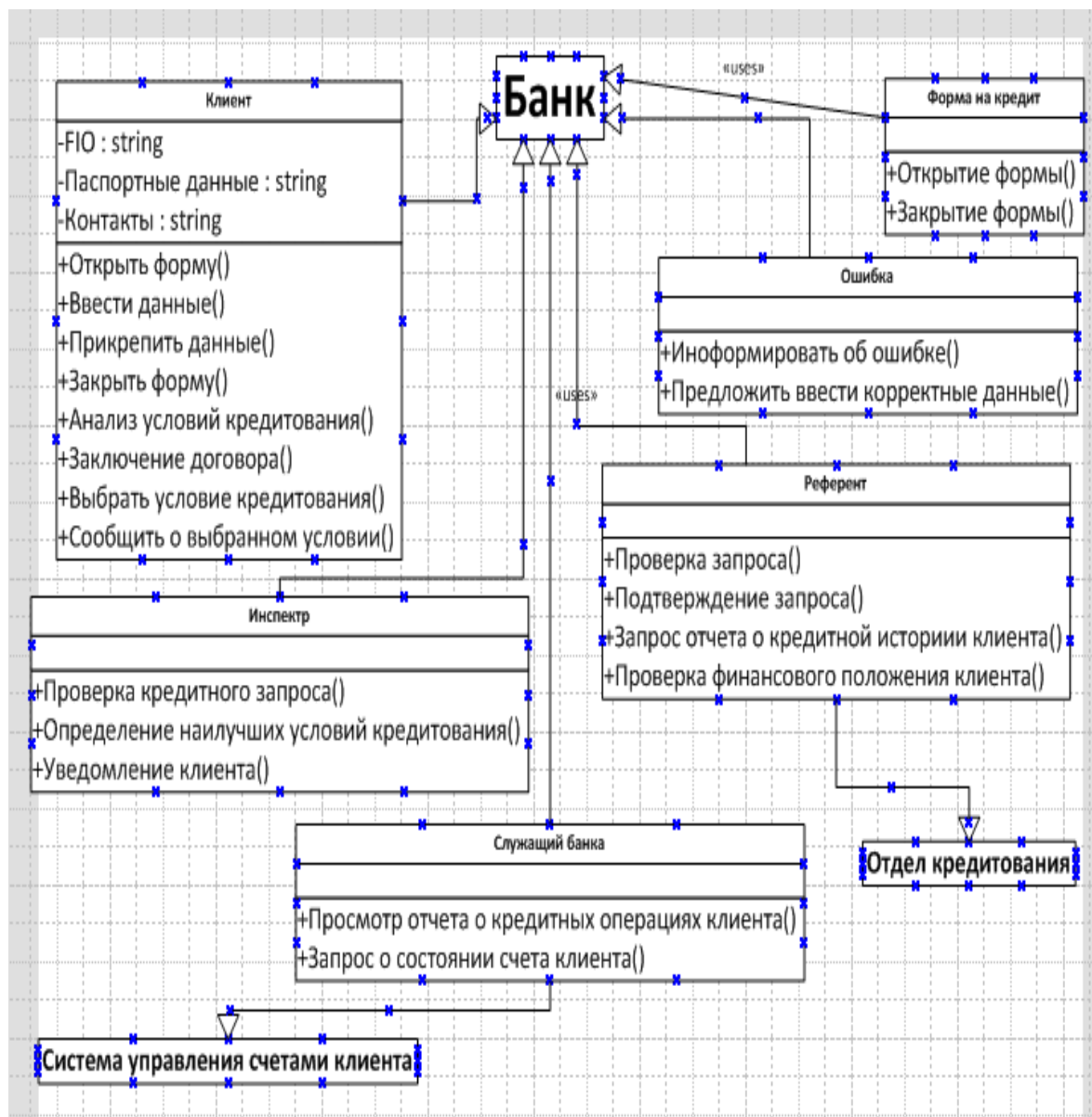


Рисунок 7 – Диаграмма классов

1.4 Теоретические основы организации кредитования ПАО «Сбербанк России»

1.4.1 Функции, принципы и формы кредита

Кредит – это сделка, заключаемая между кредитной организацией и заемщиком, по предоставлению денежных средств кредитором заемщику в виде ссуды на определённых условиях. Между кредитором и заемщиком заключается договор, согласно которому кредитор предоставляет денежные средства (кредит) заемщику в заранее размере суммы, на оговоренный срок, и за вознаграждение кредитора. Кредит может предоставляться с обеспечением, с поручительством или без него, все это рассматривается кредитной организацией в каждом конкретном случае. Кредит предоставляется на различные цели и на различных условиях, как для физических лиц, так и для бизнеса. Кредит может предоставляться, например, на покупку автомобиля, недвижимости, бытовой техники и прочих товаров и услуг, а также на любые другие цели [17].

Субъектами кредитных отношений являются кредитор и заемщик. Ими могут быть любые юридически самостоятельные лица и дееспособные граждане, которые в состоянии нести материальную ответственность по обязательствам кредитной сделки.

Кредитор – это субъект кредитных отношений, передающий стоимость во временное пользование, а заемщик - субъект, получающий кредит и обязанный вернуть его в установленный срок. В рамках кредитных отношений они могут меняться ролями: кредитор может стать заемщиком, а заемщик - кредитором. Для современного уровня развития товарно-денежных отношений характерно также одновременное функционирование субъектов в качестве и кредиторов, и заемщиков. Так,

например, банки в одно и то же время на протяжении всей своей деятельности являются и кредиторами, и заемщиками [17].

Объектом кредитной сделки выступает ссуженная стоимость, то есть стоимость в денежной или товарной форме, которую кредитор передает во временное пользование заемщику.

Основные принципы кредита:

— Возвратность кредита. Этот принцип выражает необходимость своевременного возврата полученных от кредитора финансовых ресурсов после завершения их использования заемщиком. Он находит свое практическое выражение в погашении конкретной ссуды путем перечисления соответствующей суммы денежных средств на счет предоставившей ее кредитной организации.

— Срочность кредита. Этот принцип отражает необходимость его возврата не в любое приемлемое время для заемщика, а в точно определенный срок, зафиксированный в кредитном договоре. Сроки кредитования устанавливаются банком, исходя из сроков оборачиваемости кредитруемых материальных ценностей и окупаемости затрат, но не выше нормативных.

— Платность кредита. Этот принцип выражает необходимость не только прямого возврата заемщиком полученных от банка кредитных ресурсов, но и внесения банку определенной платы за пользование ссудой. Экономическая сущность платы за кредит отражается в фактическом распределении полученной предприятием прибыли за счет использования ссуды между банком и предприятием.

— Обеспеченность кредита. Этот принцип выражает необходимость обеспечения защиты имущественных интересов кредитора при возможном нарушении заемщиком принятых на себя обязательств. Т.е. наличие у заемщика юридически оформленных обязательств, гарантирующих своевременный возврат кредита: залогового обязательства,

договора-гарантии, договора-поручительства, договора-страхования ответственности непогашения кредита.

— Дифференцированность кредитования означает, что коммерческие банки не должны однозначно подходить к вопросу о выдаче кредита своим клиентам, претендующим на получения кредита. Кредит должен предоставляться только тем предприятиям, которые в состоянии вернуть его своевременно. Дифференцированность кредитования должна осуществляться на основании показателей кредитоспособности, под которой понимается финансовое состояние предприятия, дающее уверенность в способности и готовности заемщика возратить кредит в обусловленный договором срок.

Формы кредита:

— Банковский кредит одна из наиболее распространенных форм кредитных отношений в экономике, объектом которых выступает процесс передачи в ссуду денежных средств. Банковский кредит представляется, исключительно, кредитно-финансовыми организациями, имеющими лицензию на осуществление подобных операций от Центрального Банка. В роли заемщика выступают юридические лица, инструментом кредитных отношений является кредитный договор. Доход по этой форме кредита банк получает в виде ссудного процента или банковского процента.

— Международный кредит носит как частный, так и государственный характер, отражая движение ссудного капитала в сфере международных экономических и валютно-финансовых отношений.

— Коммерческий кредит, можно охарактеризовать как кредит, предоставляемый в товарной форме продавцами покупателям в виде отсрочки платежа за проданные товары. Он предоставляется под обязательства должника (покупателя) погасить в определенный срок, как сумму основного долга, так и начисляемые проценты.

— Потребительский кредит, как правило, предоставляется торговыми компаниями, банками и специализированными кредитно-финансовыми институтами для приобретения населением товаров и услуг с рассрочкой платежа.

— Ипотечный кредит выдается на приобретение либо строительство жилья, либо покупку земли. Предоставляют его банки (кроме инвестиционных) и специализированные кредитно-финансовые институты.

— Государственный кредит следует разделять на собственно государственный кредит и государственный долг. В первом случае кредитные институты государства (банки и другие кредитно-финансовые институты) кредитуют различные секторы экономики. Во втором случае государство заимствует денежные средства у банков и других кредитно-финансовых институтов на рынке капиталов для финансирования бюджетного дефицита и государственного долга. При этом государственные облигации покупают население, юридические лица, различные предприятия и компании.

— Международный кредит носит как частный, так и государственный характер, отражая движение ссудного капитала в сфере международных экономических и валютно-финансовых отношений [12].

1.5 Постановка цели работы. Формулировка задач, подлежащих решению

Целью работы является улучшение качества обслуживания клиентов Абаканского отделения №8602 Сбербанка России за счет разработки потребительского калькулятора. Данный кредитный калькулятор позволит облегчить работу специалиста кредитного отдела, уменьшив объем времени, потраченного на оказание услуги и на расчет выбранного клиентом вида кредита.

Для достижения цели необходимо решить следующие задачи:

- Проанализировать деятельность кредитного отдела.
- Построить диаграмму вариантов использования информационной системы отдела кредитования в Абаканском отделении №8602 Сбербанка России.
- Разработать калькулятор для потребительского кредита Абаканскому отделению №8602 «Сбербанка России».
- Рассчитать экономический эффект от внедрения разработки калькулятора и ее реализации.

Разработанный кредитный калькулятор сэкономит время работы специалиста кредитного отдела, также потребителя. Калькулятор для потребительского кредита предусмотрен физическим лицам, и предусмотрена простота пользования калькулятором, он имеет необходимые функции для расчета кредита, также для улучшения работы продуманы подсказки.

1.6 Анализ литературы и других источников информации по разработке кредитного калькулятора

Мобильный калькулятор для платформ Android и iOS.

Единственный мобильный калькулятор «Универсальный калькулятор» с функцией «Потребительский кредит» (рисунок 8) и другими дополнительными формами кредитования.

Рисунок 8 – Главная форма калькулятора «Универсальный калькулятор»

«Универсальный калькулятор» позволяет совершать моментальные расчеты суммы платежа, переплаты реальной процентной ставки или эффективной процентной ставки по потребительским, ипотечным, автокредитам, а так же во вкладах. Вводите сумму кредита и первоначальный взнос слайдером и вручную, нажав на соответствующее поле.

Как написано в описании калькулятора «Имеется ввод вручную», но не имеется поле ввода вручную, этого не предусмотрел разработчик (недостоверная информация в описании универсального калькулятора).

Также имеющиеся мобильные калькуляторы «Кредитный калькулятор», «Онлайн сбербанк» и «Мобильный банк» платформа iOS и Android не могут работать без доступа в интернет.

Калькулятор затрат на потребительский кредит для стационарного компьютера.

Калькулятор потребительского кредита показан на рисунке 9, учитывает ряд дополнительных расходов, которые могут серьезно увеличить затраты по займу.

Это особенно актуально для потребительского кредита. Поскольку такие ссуды являются самыми рискованными из-за того, что не содержат дополнительного обеспечения, банки вынуждены компенсировать свои риски повышенной доходностью по этим кредитам. На открытое повышение процентной ставки готовы далеко не все банки, многие предпочитают завуалировать ее в виде различного рода комиссий.

На сегодняшний день потребительские кредиты Сбербанка России популярны в России и довольно удобны с точки зрения процентных ставок и переплат. Заполнив требуемые графы, клиент получает точный расчёт будущих ежемесячных платежей. Кредиты наличными для физических лиц от Сбербанка России доступны людям с широким диапазоном подтверждённого дохода — главное, правильно рассчитать платёж с помощью онлайн-калькулятора потребительского кредита и знать свои финансовые возможности.

Сумма	100 000,00		
Срок	18	месяцев	Дата получения 06.06.2016
Ставка	Фиксированная	12,50	% в год
Тип платежей	Аннуитентные <input type="checkbox"/> Расширенные настройки		

Название комиссии	Страхование жизни и трудоспособности [0.2 - 2%]	Ежегодно	<input type="button" value="X"/>
%	0,20	% от остатка задолженности, увеличенной на	10,00 %

Название комиссии	Комиссия за рассмотрение заявки на получение кредита [0 -	Единоновременно	<input type="button" value="X"/>
Сумма	1 000,00	рублей	

Название комиссии	Комиссия за обналичивание кредитных средств [0 - 3%]	Единоновременно	<input type="button" value="X"/>
%	1,00	% от суммы кредита	

Название комиссии	Комиссия за обслуживание и ведение расчетного счета [0 -	Ежегодно	<input type="button" value="X"/>
Сумма	1 000,00	рублей	

☐ Учитывать инфляцию

Рисунок 9– Онлайн калькулятор потребительского кредита
веб-приложении

У Сбербанка и других банков не имеется потребительский калькулятор для мобильных телефонов.

Имеющиеся калькуляторы предусмотрены для специалистов или людям, которые разбираются в кредитовании. Такими калькуляторами трудно пользоваться простым пользователям. Также все веб-калькуляторы работают с доступом интернет.

1.7 Обоснование выбора средств для разработки приложений

В кредитном отделе Абаканского отделения №8602 имеется потребительский калькулятор, недостатком которого является отсутствие

расчета пени за просроченный платеж, а также отсутствует расчет суммы кредита и срока кредита. Поэтому было решено разработать стационарный кредитный калькулятор, который решит данные проблемы.

В качестве среды разработки для мобильного приложения был выбран Android Studio и Delphi 7 для ПК.

Android Studio

Android Studio — это интегрированная среда разработки (IDE) для работы с платформой Android. Android Studio является официальным средством разработки Android приложений от компании Google.

Android Studio позволяет вам увидеть любые визуальные изменения, которые вы производите в реальном времени в приложении. Вы также можете увидеть, как ваше приложение будет одновременно смотреться на различных устройствах под управлением Android, с различными настройками и разрешением экрана.

Преимущества Android Studio:

- Легко проверить производительность приложения на различных типах устройств.
- Помощники и шаблоны для общих элементов программирования для Android.
- Полнофункциональный редактор с множеством дополнительных инструментов, способствующих ускорению разработки приложений

Языком программирования для Android Studio был выбран JavaScript.

JavaScript — это встроенный, мультиплатформенный, интерпретируемый язык, созданный компанией Netscape. Как и HTML, он совместим со всеми платформами.

Основные области использования языка JavaScript при создании интерактивных HTML-страниц:

- 1) Динамическое создание документа с помощью сценария.

2) Оперативная проверка достоверности заполняемых пользователем полей форм HTML до передачи их на сервер.

3) Создание динамических HTML-страниц совместно с каскадными таблицами стилей и объектной моделью документа. С помощью языка CSS оформляется внешний вид веб-страницы.

4) Взаимодействие с пользователем при решении “локальных” задач, решаемых приложением JavaScript, встроенном в HTML-страницу.

Существует большое количество JavaScript библиотек, каждая из которых предназначена для упрощения программирования на JavaScript. Для данного приложения использовались библиотеки jQuery и ZeroClipboard.

Как средство описания, оформления внешнего вида веб-страницы, написанной на HTML, использовался язык CSS.

CSS (англ. CascadingStyleSheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Для экономии времени и создания гармоничного дизайна использовалось средство Bootstrap — свободный набор инструментов для создания сайтов и веб-приложений [11].

Delphi 7.

Delphi — императивный, структурированный, объектно-ориентированный язык программирования со строгой статической типизацией переменных. Основная область использования — написание прикладного программного обеспечения.

Преимущества Delphi по сравнению с аналогичными программными продуктами.

- быстрота разработки приложения;
- высокая производительность разработанного приложения;
- низкие требования разработанного приложения к ресурсам компьютера;

- наращиваемость за счет встраивания новых компонент и инструментов в среду Delphi;
- возможность разработки новых компонент и инструментов собственными средствами Delphi (существующие компоненты и инструменты доступны в исходных кодах);
- удачная проработка иерархии объектов.

Система программирования Delphi рассчитана на программирование различных приложений и предоставляет большое количество компонентов для этого.

1.8 Выводы по разделу «Моделирование деятельности организации ПАО «Сбербанк России»»

Для наглядности построена диаграмма вариантов использования информационной системы отдела кредитования, где представлена информация, как происходит кредитование физическим и юридическим лицам (индивидуальных клиентов и организаций). Выявляется проблема с расчетом сумм кредитов для потребителей. Проводится анализ используемых программных продуктов в отделе кредитования. Дано обоснование собственной разработки кредитного калькулятора для потребителей (мобильный калькулятор) и разработки программного продукта для сотрудников кредитного отдела (кредитный калькулятор для ПК).

Для разработки мобильного калькулятора использован язык веб – программирования JavaScript, объектом приложения выбрана портативная операционная система Android. Для разработки кредитного калькулятора для ПК выбран объектно-ориентированный язык программирования Delphi7.

Разработанные калькуляторы для потребительского кредита будут иметь название «Потребительский калькулятор».

2 Разработка кредитного калькулятора для Абаканского отделения №8602 Сбербанка России

2.1 Функциональная архитектура «Потребительский калькулятор»

Для разработки кредитных калькуляторов создана концептуальная модель в виде диаграммы потоков данных (DFD). Для построения диаграммы использовалась программа Microsoft Office Visio. На рисунке 10 представлена начальная контекстная диаграмма разработанного потребительского калькулятора.

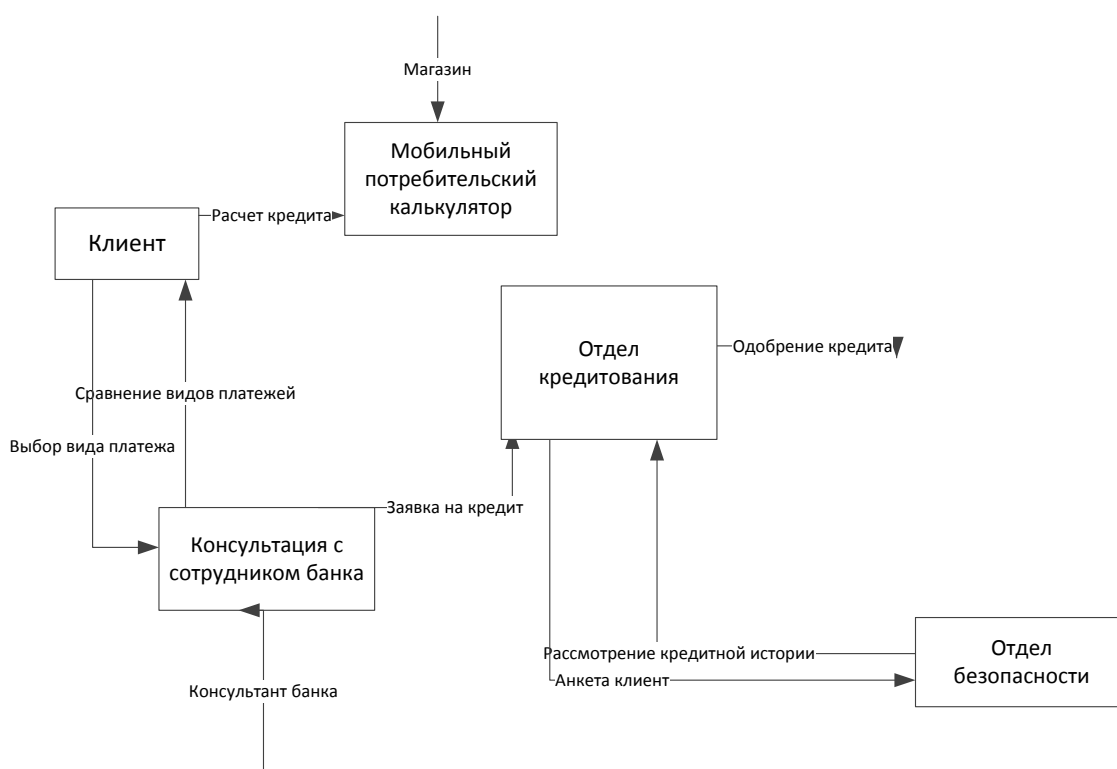


Рисунок 10 – Начальная контекстная диаграмма программы «Потребительский калькулятор»

Внешние сущности:

Клиент – это юридическое или физическое лицо, пользующееся услугами банка. Он может рассчитать сумму ежемесячного платежа с помощью мобильного потребительского калькулятора, при этом находясь в любом месте, например, в крупном магазине.

Мобильный потребительский калькулятор – это приложение, помогающее подобрать виды кредитования для расчета кредита. Он доступен для пользователей устройств на базе Android.

Отдел кредитования рассматривает всевозможные варианты для выдачи кредита, чтобы увеличить оборот выданного кредита для получения большей прибыли.

Отдел безопасности занимается проверкой документов, предотвращением мошеннических действий с кредитами, с финансовыми потоками банка, охраной здания банка и защитой информационных данных.

Консультация с сотрудниками банка. Здесь решается вопрос о том, какой выгодный кредит выдать клиенту, а также рассчитывается ежемесячные выплаты по кредиту.

2.2 Файловая структура мобильного и стационарного калькулятора

На рисунке 11 показана файловая структура веб – приложения.

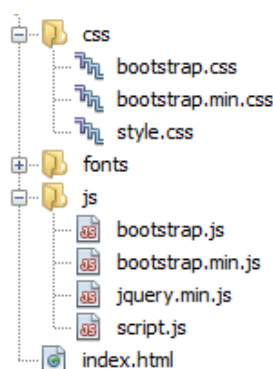


Рисунок 11 – Структура файлов веб-приложения

index.html - содержит разметку страницы, элементы управления и подключает к себе файлы .css и .js.

Файл style.css содержит CSS-стили оформления элементов html-страницы.

Файл script.js содержит JavaScript код, отвечающий за интерактивность html - элементов, а также за расчет кредитов и пеней.

Также подключены файлы фреймворка Bootstrap и библиотеки jQuery.

Структура файлов проекта из Android Studio показана на рисунке 12.

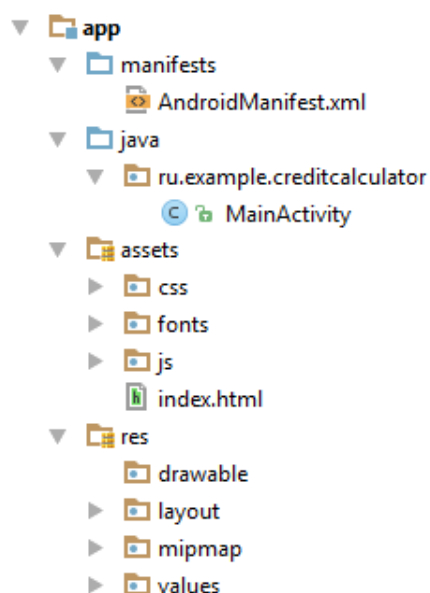


Рисунок 12 – Структура каталогов Android Studio

Файл AndroidManifest.xml содержит настройки мобильного приложения, список необходимых разрешений для обращения к защищенным частям API и взаимодействия с другими приложениями, а также минимальный и максимальный уровень OS Android, необходимый для работы приложения.

Файл MainActivity.java содержит код формы приложения, в частности код, выполняемый при событии create, инициализирующий форму и настраивающий отображение компонента WebView.

Каталог assets содержит файлы разработанной веб-страницы: index.html, стили и скрипты.

Каталог res – это папка ресурсов мобильного приложения. В частности содержит файлы иконки приложения.

Структура стационарного приложения включает файлы, представленные на рисунке 13.

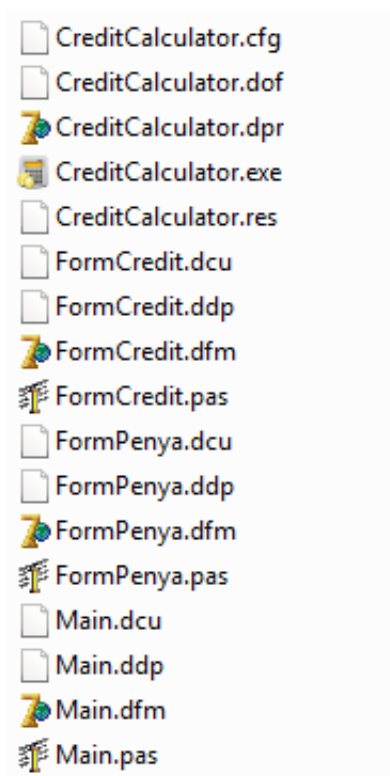


Рисунок 13 – Структура файлов приложения для ПК

Описание файлов проекта для ПК показано в таблице 2.

Таблица 2 – Файлы проекта

Имя файла	Описание
CreditCalculator.dpr	Файл проекта для хранения информации о формах и модулях. В нем содержатся операторы инициализации и запуска программ на выполнение.
FormPenya.pas	Текстовый файл модуля, используемый для хранения кода. Можно создавать модули, не связанные с формами. Многие из функций и процедур Delphi хранятся в модулях.
FormPenya.dfm	Файл формы. Двоичный или текстовый файл, который создается Delphi для хранения информации о ваших формах. Каждому файлу формы соответствует файл модуля
CreditCalculator.dof	Файл параметров проекта.
CreditCalculator.res	Файл ресурсов. Бинарный файл содержит используемую проектом пиктограмму и прочие ресурсы.
FormCredit.dfm, FormCredit.pas, FormPenya.ddp	Это соответственно файлы резервных копий для файлов проекта, формы, модуля и графических схем. Если что-то безнадежно испорчено в проекте, можно соответственно изменить расширения этих файлов и вернуться к предыдущему варианту
CreditCalculator.exe	Исполняемый файл приложения. Он является автономным.
CreditCalculator.cfg	Файл, где хранятся настройки для консольного варианта компилятора.

Окончание таблицы 2

Имя файла	Описание
FormCredit.dcu	Откомпилированный объектный файл модуля (.pas), который компонуется в окончательный исполняемый файл.
FormCredit.ddp	Файл, хранящий графические схемы, наглядно поясняющие взаимосвязи между компонентами

Выше описаны имена файлов для стационарного приложения «Потребительский калькулятор», разработанного для специалистов отдела кредитования банка.

2.3 Описание программных модулей

Программный модуль для мобильного телефона.

С помощью графического интерфейса программного средства Android Studio создан калькулятор для расчета потребительского кредита в виде веб-приложения.

Приложение «Потребительский калькулятор» запускает файл index.html, код которого представлен в приложении А. Здесь необходимо установить настройки подключения скриптов:

```
<script src="js/jquery.min.js"></script> // библиотека JQuery  
<script src="js/bootstrap.min.js"></script> // библиотека Bootstrap  
<script src="js/script.js"></script> //подключение script.js
```

Далее представлен код функции файла script.js, вычисляющей сумму пеней. Полный код данного файла представлен в приложении Б.

```

function calculatePenya() {
    var summa_zadolzhen = parseFloat($('#summa-zadolzhen').val());
    var data_platezha = new Date(Date.parse($('#data-platezha').val()));
    var data_podscheta = new Date(Date.parse($('#data-podscheta').val()));
    var penya = parseFloat($('#penya').val());

    var kolich_dnei = Math.abs(data_podscheta.getTime() - data_platezha.getTime());
    kolich_dnei = Math.ceil(kolich_dnei / (1000 * 60 * 60 * 24));

    var summa_penei = summa_zadolzhen * penya / 100 * kolich_dnei;

    $('#js-penya-result').html(
        '<div class="well well-sm well-result">' +
            '<div>Сумма задолженности: <span class="pull-right">' + summa_zadolzhen + '
руб.</span></div>' +
            '<div>Дата платежа по договору: <span class="pull-right">' +
getUserFormattedDate(data_platezha) + '</span></div>' +
            '<div>Дата погашения: <span class="pull-right">' +
getUserFormattedDate(data_podscheta) + '</span></div>' +
            '<div>Количество дней просрочки: <span class="pull-right">' + kolich_dnei +
'</span></div>' +
            '<div>Размер пени в день: <span class="pull-right">' + penya + '%</span></div>'
        +
            '<div>Сумма пеней: <span class="pull-right"><strong>' +
summa_penei.toFixed(2) + ' руб.</strong></span></div>' +
            '</div>'
    );
}

```

Программный модуль для персонального компьютера.

Полный код программного модуля main.pas представлен в приложении Г. В нем прописаны все функции разработанного стационарного приложения.

Для представления результатов расчёта пеней написан следующий код (полный код представлен в приложении Д):

```
procedure TfrmPenyaResult.btnCloseClick(Sender: TObject);
begin
    frmPenyaResult.Close;
end;

procedure TfrmPenyaResult.btnPrintClick(Sender: TObject);
begin
    frmPenyaResult.Print;
end;

end.
```

Далее представлен код для представления результатов расчёта аннуитентного и дифференцированного платежей (полный код представлен в приложении Е):

```
procedure TfrmCreditResult.btnCloseClick(Sender: TObject);
begin
    frmCreditResult.Close;
end;

procedure TfrmCreditResult.btnPrintClick(Sender: TObject);
begin
    frmCreditResult.Print;
end;

end.
```

2.4 Описание пользовательского интерфейса «Потребительский калькулятор»

2.4.1 Пользовательский интерфейс программы для мобильного калькулятора.

Приложение для мобильного телефона.

Главная страница приложения «Потребительский калькулятор» для мобильного телефона, созданная на платформе Android Studio, показана на рисунке 14. Программный код находится в приложении А.

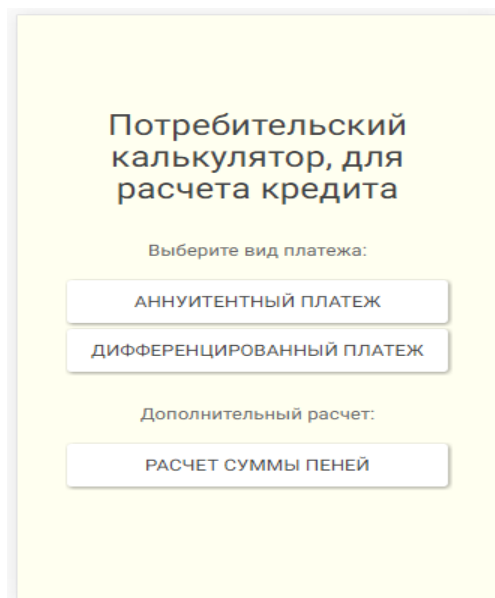


Рисунок 14 – Главная страница мобильного приложения
«Потребительский калькулятор»

Предусмотрена проверка ввода данных на некорректность. Если допустить ошибку при вводе или пропустить заполнение поля ввода, поля, в которых находятся некорректные данные, выделяются красным цветом (рисунок 15).

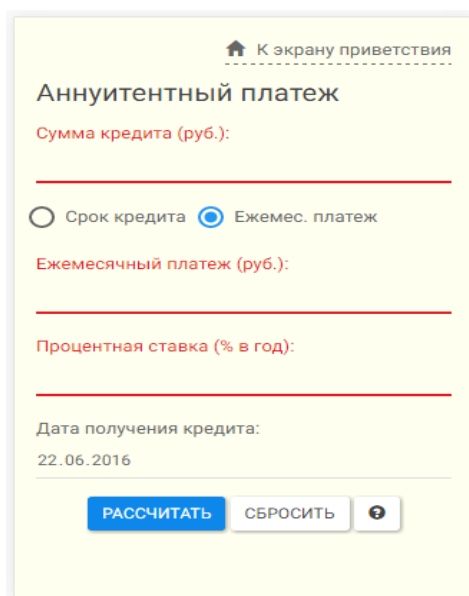


Рисунок 15 – Некорректное заполнение приложения
«Потребительский калькулятор»

На рисунке 16 показан расчет ежемесячного платежа по аннуитентному виду платежа, для расчета которого применялась формула 2. Также в этом окне имеется расчет срока кредита, если известны ежемесячный платеж, общая сумма кредита и размер процентной ставки. На форме имеется кнопка «К экрану приветствия», он направляет на главную страницу приложения.

[К экрану приветствия](#)

Аннуитентный платеж

Сумма кредита (руб.):
200000

☒ Срок кредита ☐ Ежемес. платеж

Срок кредита (мес.):
12

Процентная ставка (% в год):
14.5

Дата получения кредита:
22.06.2016

РАССЧИТАТЬ
СБРОСИТЬ
?

Сумма кредита: 200000 руб.

Срок кредита: 12 мес.

Процентная ставка: 14.5%

Дата получения кредита: 22.06.2016

Дата последней выплаты: 22.06.2017

Ежемесячный платеж: **18004.51 руб.**

Полная сумма выплат: **216054.11 руб.**

Переплата: **16054.11 руб.**

№	Дата платежа	Остаток кредита	Проценты	Осно пла
1	22.07.2016	200000.00	2416.67	1556
2	22.08.2016	184412.16	2228.31	1577
3	22.09.2016	168635.96	2037.68	1596
4	22.10.2016	152669.14	1844.75	1615
5	22.11.2016	136509.38	1649.49	1635
6	22.12.2016	120154.36	1451.87	1655
7	22.01.2017	103601.72	1251.85	1675
8	22.02.2017	86849.06	1049.43	1695
9	22.03.2017	69893.98	844.55	1715
10	22.04.2017	52734.02	637.20	1736
11	22.05.2017	35366.71	427.35	1757
12	22.06.2017	17789.55	214.96	1778

Рисунок 16 – Расчёт ежемесячного платежа по аннуитентному виду

Для удобства и простоты использования имеется кнопка в виде иконки вопросительного знака, которая при нажатии представляет небольшую справочную информацию по виду платежа (рисунок 17).

Рисунок 17 – Окно подсказки

На рисунке 18 показан расчет ежемесячного платежа по дифференциальному виду платежа, для расчета применяли формулу 3.

Чтобы рассчитать пени применялась формула:

$$СП = \frac{СЗ \cdot РП}{100} * КДП , \quad (4)$$

где СП – сумма пеней;

СЗ – сумма задолженности;

РП – размер пени в день;

КДП – количество дней просрочки.

[К экрану приветствия](#)

Дифференцированный платеж

Сумма кредита (руб.):
185000

Срок кредита (мес.):
6

Процентная ставка (% в год):
17,5

Дата получения кредита:
22.06.2016

[РАССЧИТАТЬ](#) [СБРОСИТЬ](#) [?](#)

Сумма кредита: 185000 руб.

Срок кредита: 6 мес.

Процентная ставка: 17.5%

Дата получения кредита: 22.06.2016

Дата последней выплаты: 22.12.2016

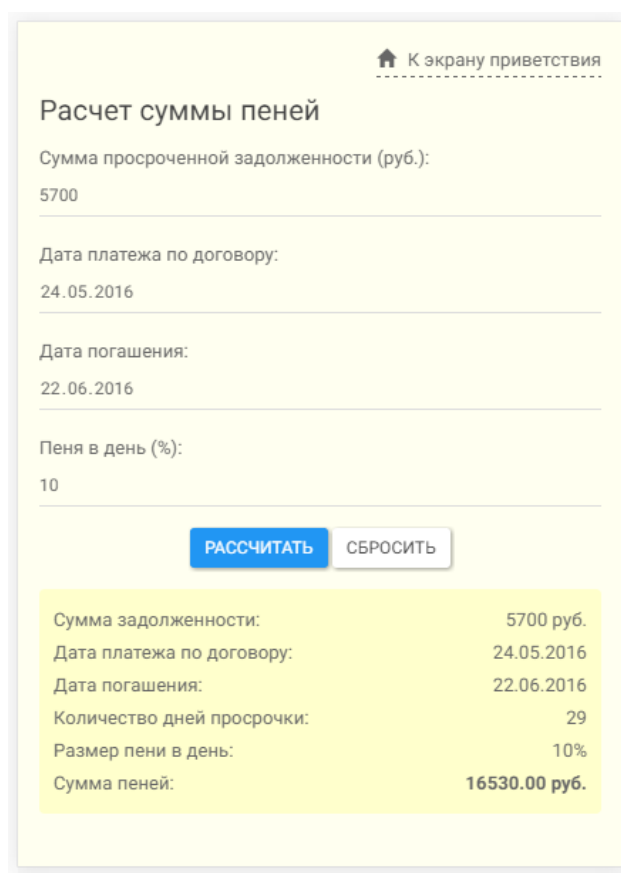
Полная сумма выплат: **194442.71 руб.**

Переплата: **9442.71 руб.**

№	Дата платежа	Остаток кредита	Проценты	Основной платеж	Всего за платеж
1	22.07.2016	185000.00	2697.92	30833.33	33531.25
2	22.08.2016	154166.67	2248.26	30833.33	33081.60
3	22.09.2016	123333.33	1798.61	30833.33	32631.94
4	22.10.2016	92500.00	1348.96	30833.33	32182.29
5	22.11.2016	61666.67	899.31	30833.33	31732.64
6	22.12.2016	30833.33	449.65	30833.33	31282.99

Рисунок 18 – Расчёт ежемесячного платежа по дифференцированному виду

Расчет пени показан на рисунке 19.



К экрану приветствия

Расчет суммы пеней

Сумма просроченной задолженности (руб.):
5700

Дата платежа по договору:
24.05.2016

Дата погашения:
22.06.2016

Пеня в день (%):
10

РАССЧИТАТЬ СБРОСИТЬ

Сумма задолженности:	5700 руб.
Дата платежа по договору:	24.05.2016
Дата погашения:	22.06.2016
Количество дней просрочки:	29
Размер пени в день:	10%
Сумма пеней:	16530.00 руб.

Рисунок 19 – Расчёт пени

Программа для персонального компьютера.

На главной форме стационарного приложения «Потребительский калькулятор» размещена информация о деталях оплаты кредита по двум видам платежа (рисунок 20).

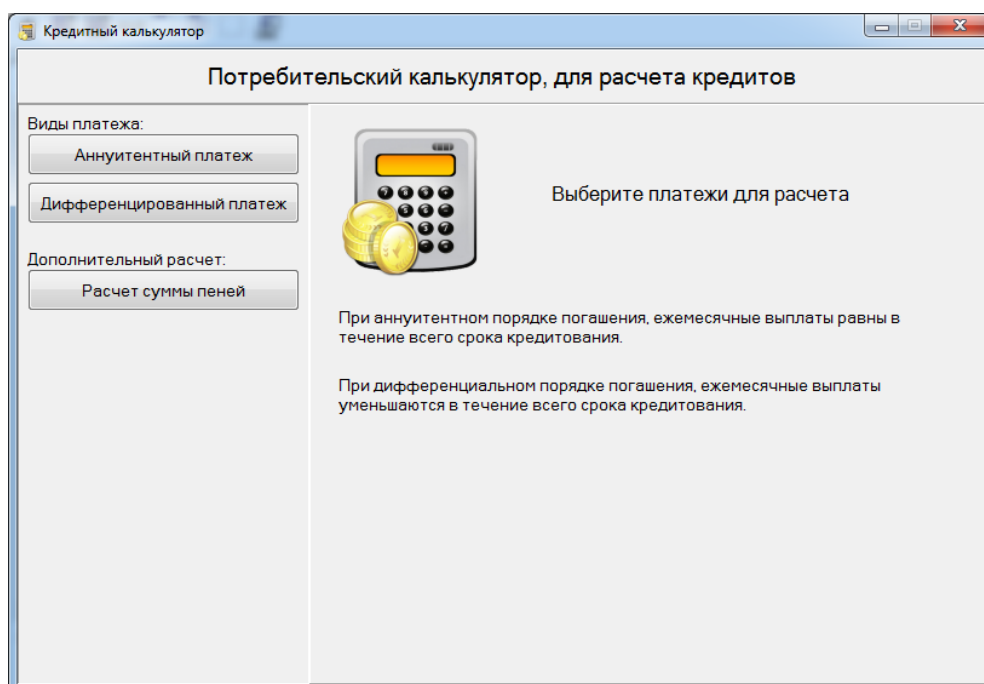


Рисунок 20 – Главная форма приложения «Потребительский калькулятор»
для ПК

Форма для расчета платежа по аннуитентному виду представлена на рисунке 22, здесь имеется дополнительная функция для расчета срока кредита.

Рисунок 22 – Расчёт ежемесячный платеж по аннуитентному виду платежа

Результаты данного расчета, а именно информация о платеже кредита и график платежей, показаны на рисунке 23. Специалист может рассчитать ежемесячный платеж или срок оплаты кредита и распечатать с помощью кнопки «Печать».

Аннуитентный платеж

Аннуитентный платеж

Сумма кредита: 150 000,00 руб.
Срок кредита: 10 мес.
Процентная ставка: 21,5%
Дата получения кредита: 21.06.2016
Дата последней выплаты: 21.04.2017
Полная сумма выплат: **165 174,77 руб.**
Переплата: **15 174,77 руб.**
Ежемесячный платеж: **16 517,48 руб.**

№	Дата платежа	Остаток кредита	Проценты	Основной платеж	Всего за платеж
1	21.07.2016	150 000,00	2 687,50	13 829,98	16 517,48
2	21.08.2016	136 170,02	2 439,71	14 077,76	16 517,48
3	21.09.2016	122 092,26	2 187,49	14 329,99	16 517,48
4	21.10.2016	107 762,27	1 930,74	14 586,74	16 517,48
5	21.11.2016	93 175,53	1 669,39	14 848,08	16 517,48
6	21.12.2016	78 327,45	1 403,37	15 114,11	16 517,48
7	21.01.2017	63 213,34	1 132,57	15 384,90	16 517,48
8	21.02.2017	47 828,43	856,93	15 660,55	16 517,48
9	21.03.2017	32 167,88	576,34	15 941,14	16 517,48
10	21.04.2017	16 226,75	290,73	16 226,75	16 517,48

Печать

Заккрыть

Рисунок 23 – Результаты расчета ежемесячного платежа

Также показан расчет по дифференцированному виду платежа на рисунке 24.

Кредитный калькулятор

Потребительский калькулятор, для расчета кредитов

Виды платежа:

Дополнительный расчет:

Дифференцированный платеж

Сумма кредита (руб.): 120000

☒ Срок кредита ☐ Ежемесячный платеж

Срок кредита (мес.): 10

Процентная ставка (% в год): 19,5

Дата получения кредита: 22.06.2016

Рисунок 24 – Расчёт ежемесячного платежа по дифференцированному виду платежа

Информация о платеже кредита и график платежей показаны на рисунке 25. Также предусмотрена функция для печати графика платежей.

Дифференцированный платеж

Сумма кредита: 120 000,00 руб.
Срок кредита: 10 мес.
Процентная ставка: 19,5%
Дата получения кредита: 22.06.2016
Дата последней выплаты: 22.04.2017
Полная сумма выплат: **128 775,00 руб.**
Переплата: **8 775,00 руб.**

№	Дата платежа	Остаток кредита	Проценты	Основной платеж	Всего за платеж
1	22.07.2016	120 000,00	1 950,00	12 000,00	13 950,00
2	22.08.2016	108 000,00	1 755,00	12 000,00	13 755,00
3	22.09.2016	96 000,00	1 560,00	12 000,00	13 560,00
4	22.10.2016	84 000,00	1 365,00	12 000,00	13 365,00
5	22.11.2016	72 000,00	1 170,00	12 000,00	13 170,00
6	22.12.2016	60 000,00	975,00	12 000,00	12 975,00
7	22.01.2017	48 000,00	780,00	12 000,00	12 780,00
8	22.02.2017	36 000,00	585,00	12 000,00	12 585,00
9	22.03.2017	24 000,00	390,00	12 000,00	12 390,00
10	22.04.2017	12 000,00	195,00	12 000,00	12 195,00

Рисунок 25 – Результаты расчета ежемесячного платежа

Специалист отдела кредитования может рассчитать пени (рисунок 26) клиента (применена формула 4) и информировать клиента о задолженности пеней.

Кредитный калькулятор

Потребительский калькулятор, для расчета кредитов

Виды платежа:

Дополнительный расчет:

Расчет суммы пеней

Сумма просроченной задолженности (руб.):

3500

Дата платежа по договору:

22.05.2016

Дата платежа погашения:

22.06.2016

Пеня в день (%):

5,5

Рисунок 26 – Расчет пени

Результат расчета пени можно вывести на печать (рисунок 27).

Сумма пеней

Расчет суммы пеней:

Сумма задолженности:	3 500,00 руб.
Дата платежа по договору:	22.05.2016
Дата погашения:	22.06.2016
Количество дней просрочки:	31
Размер пени в день:	5,5%
Сумма пеней:	5 967.50 руб.

Рисунок 27 – Результат расчета сумма пеней

2.5 Вывод по разделу «Разработка кредитного калькулятора для Абаканского отделения №8602 Сбербанка России»

Построена начальная контекстная диаграмма программы «Потребительский калькулятор».

На основе диаграммы потоков данных выявлены основные сущности, взаимодействующие с приложениями.

Для разработанного мобильного приложения «Потребительский калькулятор» написан программный код на языке программирования JavaScript, при использовании и других средств веб-программирования, включая среду разработки Android Studio.

Для разработанного стационарного приложения «Потребительский калькулятор» использовалась интегрированная среда разработки Delphi 7.

Результатная информация работы приложений представляет данные в табличном виде о расчетах платежей по выбранному виду кредитования.

Созданные приложения для расчета потребительского кредита для ПАО «Сбербанк России» позволят сократить временные затраты за счет оперативной работы с удобным и понятным интерфейсом разработанных программных продуктов.

3 Расчет показателей экономической эффективности от внедрения потребительского калькулятора в Абаканском отделении №8602 Сбербанка России

3.1 Расчет трудозатрат на разработку программного продукта «Потребительский калькулятор»

Трудозатраты измеряются в чел.*час. Расчет производится по формуле:

$$T = t_{\text{и}} + t_{\text{а}} + t_{\text{п}} + t_{\text{отл}} + t_{\text{д}}, \quad (5)$$

где $t_{\text{и}}$ – затраты труда на изучение задачи;
 $t_{\text{а}}$ – затраты на разработку модели системы;
 $t_{\text{п}}$ – затраты на программирование;
 $t_{\text{отл}}$ – затраты на отладку программы;
 $t_{\text{д}}$ – затраты на подготовку документации.

Затраты труда на изучение задачи с учетом уточнения описания и квалификации программиста вычисляются по формуле:

$$t_{\text{и}} = \frac{Q \cdot B}{75 \dots 85} \cdot k, \quad (6)$$

где Q – условное число операторов в программе;
 B – коэффициент увеличения затрат в зависимости от сложности программы (1,2...5);
 K – коэффициент квалификации разработчика.

Составляющие затраты труда можно определить через условное число операторов в программном продукте. В их число входят те операторы, которые нужно учесть программисту в процессе работы над задачей с

учетом возможных уточнений постановки задачи и совершенствования алгоритма.

$$Q = q \cdot c \cdot (1 + p), \quad (7)$$

где q – предполагаемое число операторов;

c – коэффициент сложности программы (от 1 до 2);

p – коэффициент коррекции программы в ходе ее разработки (от 0,5 до 1).

Для расчета затрат следует применить усредненные значения:

$$q = 900;$$

$$c = 1,8;$$

$$p = 0,7.$$

Условное число операторов составляет:

$$Q = 900 \cdot 1,8 \cdot (1 + 0,7) = 2\,754 \text{ операторов.}$$

Коэффициент увеличения затрат характеризует увеличение затрат труда вследствие недостаточно полного описания задачи, уточнений и некоторой доработки. Этот коэффициент может принимать значения от 1,2 до 5. Необходимо взять среднее для наиболее точных расчетов: $B = 2$.

Коэффициент квалификации разработчика зависит от стажа работы программиста следующим образом:

$$\text{стаж до 2 лет} - k = 0,8;$$

$$\text{от 2 до 3 лет} - k = 1;$$

$$\text{от 3 до 7 лет} - k = 1,3 \dots 1,4;$$

$$\text{от 7 лет} - k = 1,5 \dots 1,6.$$

Так как предусмотрен небольшой набор операторов, можно будет нанимать разработчика со средним опытом работы: $k = 1$ (стаж работы от 2 до 3 лет).

Затраты труда на изучение задачи составят:

$$t_{\text{и}} = \frac{(2\,754 \cdot 2)}{75 \cdot 1} = 73,44 \text{ чел} \cdot \text{час}.$$

Расчет затрат на разработку модели системы производится по формуле:

$$t_{\text{а}} = \frac{Q}{60 \dots 75} \cdot k, \quad (8)$$

где Q – условное число операторов;

k – коэффициент квалификации разработчика.

Затраты на разработку модели системы составят:

$$t_{\text{а}} = \frac{2\,754Q}{65 \cdot 1} = 42,4 \text{ чел} \cdot \text{час}.$$

Далее необходимо рассчитать трудозатраты на отладку программы программистом.

Расчет трудозатрат на отладку производится по формуле:

$$t_{\text{отл}} = \frac{Q}{40 \dots 50} \cdot k \quad (9)$$

Затраты на отладку программы на компьютере составят:

$$t_{\text{отл}} = \frac{2\,754}{45 \cdot 1} = 61,2 \text{ чел} \cdot \text{час}.$$

При комплексной отладке программы следует предусмотреть возрастающие в 1,5 раза затраты, поэтому окончательные трудовые затраты на отладку программы будут равны:

$$t_{\text{отл.ок}} = t_{\text{отл}} \cdot 1,5 \quad (10)$$

Затраты на окончательную отладку программы на ПК составят:

$$t_{\text{отл.ок}} = 61,2 \cdot 1,5 = 91,8 \text{ чел} \cdot \text{час.}$$

Трудовые затраты на подготовку документации включают в себя затраты труда на подготовку рукописного текста и затраты труда на редактирование, печать и оформление документации.

$$t_{\text{д}} = t_{\text{дп}} \cdot t_{\text{др}}, \quad (11)$$

где $t_{\text{дп}}$ – затраты труда на подготовку материалов в рукописи;

$t_{\text{др}}$ – затраты труда на редактирование, печать и оформление различной документации.

Трудозатраты на подготовку материалов в рукописи рассчитываются по формуле:

$$t_{\text{дп}} = Q / (150 \dots 200) \cdot k, \quad (12)$$

где Q – условное число операторов;

k – коэффициент квалификации разработчика.

Исходя из имеющихся данных, трудозатраты на подготовку материалов в рукописи составят:

$$t_{\text{дп}} = 2\,754 / (175 \cdot 1) = 15,73 \text{ чел} \cdot \text{час.}$$

Затраты на редактирование, печать и оформление $t_{тр}$ прямо пропорционально зависят от затрат на подготовку материалов в рукописи:

$$t_{др} = 0,75 \cdot t_{дп} \quad (13)$$

Результаты расчетов:

$$t_{др} = 0,75 \cdot 15,73 = 11,8 \text{ чел} \cdot \text{час.}$$

Таким образом, общие трудовые затраты на подготовку документации составят:

$$t_{др} = 15,73 + 11,8 = 27,54 \text{ чел} \cdot \text{час.}$$

Теперь необходимо провести расчет трудозатрат на программирование, то есть на написание исходного кода программы.

Затраты на программирование составляет примерно 20 – 30 % от общих трудозатрат на остальные этапы разработки программы.

Расчет затрат на программирование, как 25% от общих трудозатрат, посредством выражения из формулы (1) и подстановки известных значений:

$$t_{др} = (73,44 + 42,4 + 91,8 + 27,54) \cdot 0,25 = 58,8 \text{ чел} \cdot \text{час.}$$

Общие трудозатраты составят:

$$t_{др} = 73,44 + 42,4 + 91,8 + 27,54 + 58,8 = 293,9 \text{ чел} \cdot \text{час.}$$

Полученные трудозатраты на разработку «Калькулятор потребительский» представлены в таблице 3.

Таблица 3 – Трудозатраты на разработку «Калькулятор потребительский»

Вид трудозатрат	Трудоемкость, чел*час	Удельный вес, %
Затраты на изучение задачи	73,44	24,98
Затраты на разработку модели системы	42,4	14,42
Затраты на программирование	58,8	20
Затраты на отладку программы	61,2	31,23
Затраты на подготовку документации	27,54	9,37
Итого	263,3	100

В результате составленной таблицы наибольшее количество трудозатрат при разработке программного продукта приходится на отладку программы и изучение задачи.

3.2 Экономическая оценка разработки программного продукта «Потребительский калькулятор»

Для оценки экономического эффекта от внедрения разработки «Потребительский калькулятор» необходимо рассчитать затраты на разработку программы.

Величина затрат на разработку программы «Потребительский калькулятор» определяется на основе метода калькуляций по отдельным статьям расходов и их последующим суммированием.

Разработка программы «Потребительский калькулятор» для отдела кредитования Абаканского отделения №8602 ПАО «Сбербанк России» состоит из нескольких стадий проектирования. В качестве разработчиков системы выступают: руководитель и программист (студент). В таблице 4 представлен комплекс работ по стадиям проектирования.

Таблица 4 – Комплекс работ по стадиям проектирования программы «Потребительский калькулятор»

Стадия проектирования	Исполнители	
	Руководитель	Программист
Системный анализ	+	+
Анализ требований	+	+
Проектирование	–	+
Программирование	–	+
Тестирование	–	+
Итого	2	5

Таким образом, руководитель участвует 40% времени, а программист 100% времени в проекте. Программист принимается за единицу приведенного исполнителя, следовательно, руководитель составляет 0,4 приведенного исполнителя. Т.е. в разработке проекта участвует 1,4 приведенного человека.

$$\text{Трудоемкость} = 293,9 / 1,4 = 209,95 \text{ чел} * \text{час.}$$

Срок исполнения, учитывая, что рабочий день составляет 8 ч., получается 27 день работы над проектом.

По данным компании Rational Software в реальности длительность работы над проектом на 30% больше, поэтому можно принять за срок разработки 35 день.

Постоянные издержки

Постоянные издержки включают в себя амортизационные отчисления на компьютер и программное обеспечение и затраты на текущий ремонт.

Амортизационные отчисления

Амортизационные отчисления на компьютер и программное обеспечение производятся ускоренным методом с тем условием, что срок морального старения происходит через четыре года.

Балансовая стоимость ЭВМ включает отпускную цену, расходы на транспортировку, монтаж оборудования и его наладку и вычисляется по формуле:

$$C_{\text{бал}} = C_{\text{рын}} + Z_{\text{пр}}, \quad (14)$$

где $C_{\text{бал}}$ – балансовая стоимость ЭВМ, р.;

$C_{\text{рын}}$ – рыночная стоимость компьютера, р./шт.;

$Z_{\text{пр}}$ – прочие затраты (на доставку и установку, от 8 до 10% от стоимости ПК, р.).

В процессе разработки АИС использовался ноутбук SAMSUNG 3530EA стоимостью 22 000 рублей. $Z_{\text{пр}} = 9\%$ от рыночной стоимости ПК.

$$Z_{\text{пр}} = 22\,000 \cdot 0,09 = 1980 \text{ руб.}$$

Балансовая стоимость при этом будет составлять:

$$C_{\text{бал}} = 22\,000 + 1980 = 23\,980 \text{ руб./шт.}$$

Сумма годовых амортизационных отчислений определяется по формуле:

$$A_{\Gamma} = C_{\text{бал}} \cdot H_{\text{ам}}/100, \quad (15)$$

где A_{Γ} – сумма годовых амортизационных отчислений, р.;

$C_{\text{бал}}$ – балансовая стоимость компьютера, р./шт.;

$H_{\text{ам}}$ – норма амортизации, %.

Норма амортизации составит 25%. Итог расчета A_{Γ} :

$$A_{\Gamma_{\text{ЭВМ}}} = 23\,980 \cdot 25/100 = 5\,995 \text{ руб.}$$

Общая сумма амортизационных отчислений за период создания программ «Потребительский калькулятор» равняется произведению амортизационных отчислений в день на количество дней эксплуатации компьютера и программного.

$$A_{\text{ПЭВМ}} = \frac{5\,995 \cdot 35}{366} = 573,3 \text{ руб.}$$

Общая стоимость программного обеспечения, которое используется для разработки «Калькулятор потребительский», представлена в таблице 5.

Таблица 5– Стоимость ПО, используемого для разработки программ «Потребительский калькулятор»

Программное обеспечение	Количество	Стоимость, руб.
ОС Microsoft Windows 7 Professional	1	7 390
Microsoft Office 2010 Home and Student 2010	1	4 390
Delphi 7	1	46 000
Итого		57 780

Остальные программные средства, используемые для разработки, предоставляются бесплатно.

Сумма годовых амортизационных отчислений на программное обеспечение составляет:

$$A_{\text{ГПО}} = 57\,780 \cdot 0,25 = 14\,445 \text{ руб.}$$

Общая сумма амортизационных отчислений на программное обеспечение за период создания программ:

$$A_{\text{ПЭВМ}} = \frac{14\,445 \cdot 35}{366} = 1381,3 \text{ руб.}$$

Амортизация за время эксплуатации компьютера и программного обеспечения при проектировании и разработке программ вычисляется по формуле:

$$A_{\Pi} = A_{\text{ЭВМ}} + A_{\text{ПО}}, \quad (16)$$

где $A_{\text{ЭВМ}}$ – амортизационные отчисления на компьютер за время его эксплуатации;

$A_{\text{ПО}}$ – амортизационные отчисления на программное обеспечение за время его эксплуатации.

Таким образом, совокупные амортизационные отчисления за период разработки программ составляют:

$$A_{\Pi} = 573,3 + 1381,3 = 1\,954,6 \text{ руб.}$$

Затраты на текущий ремонт

Затраты на текущий и профилактический ремонт принимаются равными 5% от стоимости ЭВМ, вычисляются по формуле:

$$З_{\text{тр}} = \frac{C_{\text{бал}} \cdot P_r \cdot T_k}{366}, \quad (17)$$

где P_r – процент на текущий ремонт, %;

T_k – временные затраты на разработку программ, дни.

Итог расчета затрат на текущий ремонт:

$$З_{\text{тр}} = \frac{23\,980 \cdot 0,05 \cdot 35}{366} = 114,7 \text{ руб.}$$

Полученные данные составляют постоянные издержки и приведены в таблице 6. Большую долю в постоянных издержках занимают

амортизационные отчисления на используемый компьютер и программное обеспечение.

Таблица 6– Постоянные издержки

Вид постоянных издержек	Затраты, руб.	Удельный вес, %
Амортизационные отчисления	1 954,6	87,7
Текущий ремонт	114,7	12,3
Итого	2 069,3	100

Переменные издержки

Переменные издержки включают в себя:

- материальные затраты;
- затраты на потребляемую электроэнергию;
- затраты на оплату труда;
- отчисления по налогам.

Материальные затраты

К материальным затратам относятся отчисления на материалы, которые используются в процессе разработки и внедрения «Калькулятор потребительский». В стоимость затрат на материалы включаются транспортные расходы (5% от стоимости материалов).

Использованные для работы материалы представлены в таблице 7.

Таблица 6– Материальные затраты

Наименование материала	Цена, руб.	Количество	Стоимость
Бумага для печати	170	1	170
Ручка, карандаш	15	2	20
Заправка картриджа принтера	350	1	350
Флеш-накопитель 16 Гб	340	1	340
Итого			880

Затраты на материалы и покупные изделия с учетом транспортных расходов равны:

$$З_{\text{м}} = 880 \cdot 1,05 = 924 \text{ руб.}$$

Затраты на электроэнергию

К затратам на электроэнергию относится стоимость потребляемой электроэнергии ЭВМ за период разработки программ.

Стоимость электроэнергии, потребляемой за год, рассчитывается по формуле:

$$З_{\text{эл}} = P_{\text{эвм}} \cdot T_{\text{эвм}} \cdot C_{\text{эл}}, \quad (18)$$

где $P_{\text{эвм}}$ – суммарная мощность ЭВМ, кВт;

$T_{\text{эвм}}$ – время работы компьютера, часов;

$C_{\text{эл}}$ – стоимость 1 кВт/ч электроэнергии, р.

В процессе разработки программ использовался ноутбук SAMSUNG 3530EA, суммарная мощность которого равна $P_{\text{эвм}} = 480 \text{ Вт}$. Цена 1 кВт/ч электроэнергии составляет $C_{\text{эл}} = 1,84 \text{ руб.}$

Рабочий день составляет восемь часов. Исходя из представленных данных, необходимо рассчитать стоимость электроэнергии за период разработки программ «Потребительский калькулятор»:

$$З_{\text{эл}} = 0,48 \cdot 35 \cdot 8 \cdot 1,84 = 247,3 \text{ руб.}$$

Затраты на оплату труда

Затраты на оплату труда разработчиков кредитных калькуляторов включают в себя следующие компоненты:

- основная заработная плата разработчиков кредитных калькуляторов;
- дополнительная заработная плата разработчиков, которая входит в фонд заработной платы.

Расчет основной заработной платы производится на основе доли выполнения работы и величины месячного должностного оклада разработчика. Среднее количество рабочих дней в месяце равно 21.

При расчете основной заработной платы, представленном в таблице 9, за период разработки программ необходимо учесть, что руководитель участвует в разработке проекта только на этапах системного анализа и анализа требований. Расчет оклада разработчиков за один рабочий день представлен в таблице 8.

Таблица 8 – Оклад разработчиков «Калькулятор потребительский»

Должность	Оклад в месяц, руб.	Стоимость одного рабочего дня, руб.
Руководитель	34 000	1 619,05
Программист	21 000	1000

Таблица 9 – Расчет основной заработной платы

Исполнитель	Трудоемкость, дни.	Стоимость одного рабочего дня, руб.	Сумма, руб.
Руководитель	5	1 619,05	8095,2
Программист	35	1000	35000
Итого			43 095,2

В дополнительную заработную плату входит районный коэффициент равный 30% и северный коэффициент, также равный 30%.

Расчет дополнительной заработной платы производится от основной заработной платы и соответственно будет равен:

$$P = 43\,095,2 \cdot 0,6 = 25\,857,14 \text{ руб.}$$

Затраты на фонд заработной платы

Затраты на фонд заработной платы приведены в таблице 10.

Таблица 10 – Расчет фонда заработной платы за период разработки программ

Вид заработной платы	Руководитель, руб.	Программист, руб.	Сумма, руб.	Удельный вес
Основная	8 095,2	35 000	43 095,2	62,5
Дополнительная	4 857,14	21 000	25 857,14	37,5
Итого	12 952,4	56 000	68 952,4	100

Отчисления по налогам

Отчисления по налогам включают отчисления по единому социальному налогу (ЕСН), который составляет 30%. Затраты по данной статье определяются установленным нормативом от расходов на оплату труда работников, непосредственно занятых созданием программ «Потребительский калькулятор», приведены в таблице 11 и вычисляются по формуле:

$$З_{пн} = З_{фз} \cdot Н_{есн}, \quad (19)$$

где $З_{пн}$ – затраты по ЕСН, р.;

$З_{фз}$ – фонд заработной платы, р.;

$Н_{есн}$ – вид налоговых отчислений, %.

Таблица 11 – Отчисления по ЕСН

Исполнитель	Фонд заработной платы, руб.	Общая сумма ЕСН, руб.
Руководитель	12 952,4	3 885,7
Программист	56 000	16 800
Итого	68 952,4	20 685,7

Далее в таблице 12 приведен список переменных издержек.

Таблица 13 – Переменные издержки

Вид переменных издержек	Величина, руб.	Удельный вес, %
Материальные затраты	924	1,01
Затраты на электроэнергию	247,3	0,27
Затраты на оплату труда	68 952,4	75,9
Отчисления по налогам	20 685,7	22,8
Итого	90 809,4	100

Далее необходимо рассчитать общие затраты. На эту статью относятся все издержки, которые были произведены при создании программ.

Полная себестоимость разработки определяется суммированием постоянных и переменных издержек и вычисляется по формуле:

$$З_{об} = З_{пос} + З_{пер}, \quad (20)$$

где $З_{об}$ – себестоимость АИС;

$З_{пос}$ – постоянные издержки;

$З_{пер}$ – переменные издержки.

Таким образом, себестоимость разработанной программы «Потребительский калькулятор» равна $З_{об} = 90\,809,4$ руб.

В таблице 13 приведена структура полных издержек. В результате, при создании программного продукта наибольший удельный вес занимают переменные издержки.

Таблица 13 – Структура полных издержек

Вид издержек	Величина, руб.	Удельный вес, %
Постоянные	2 069,3	2,23
Переменные	90 809,4	97,77
Итого	92 878,7	100

В таблице 14 показано процентное соотношение всех категорий затрат на разработку программ «Потребительский калькулятор».

Таблица 14 – Соотношение категорий затрат на разработку программ.

Вид затрат	Величина, руб.	Удельный вес, %
Материальные затраты	924	0,99
Затраты на электроэнергию	247,3	0,27
Затраты на оплату труда	68 952,4	74,2
Отчисления по налогам	20 685,7	22,3
Амортизационные отчисления	1954,6	2,1
Текущий ремонт	114,7	0,12
Итого	92 878,7	100

3.3 Расчет эффективности внедрения разработанных программ

Значение экономической эффективности от использования программ по формуле:

$$\mathcal{E} = \mathcal{Z}_6 - \mathcal{Z}_3, \quad (21)$$

где \mathcal{Z}_6 – затраты по базовому варианту (руб./год);

\mathcal{Z}_3 – затраты при использовании программного обеспечения (руб./год).

Затраты по базовому варианту рассчитываются по формуле:

$$\mathcal{Z}_6 = \text{СЧ}_\text{п} \cdot T_\text{р} \cdot (1/d_{\text{зп}}), \quad (22)$$

где $\text{СЧ}_\text{п}$ – часовая средняя заработная плата (510 руб./час);

T_p – трудоёмкость решаемой задачи;

$d_{3П}$ – доля заработной платы в общих затратах организации (74,2%).

Для решения задачи без использования разрабатываемого программного продукта необходимо примерно 40% действующего фонда рабочего времени ЭВМ, который рассчитывается по формуле:

$$T_{пк} = N_m \cdot N_d \cdot N_{\text{ч}}, \quad (23)$$

где N_m – количество месяцев в году (12);

N_d – количество рабочих дней в месяце (21);

$N_{\text{ч}}$ – средняя продолжительность рабочего дня (8 часов).

Действительный годовой фонд времени ЭВМ равен:

$$T_{пк} = 12 \cdot 21 \cdot 8 = 2016 \text{ часов/год}$$

Расчет 40% от результата:

$$T_p = 2016 \cdot 0,4 = 806,4 \text{ часов/год}$$

Затраты по базовому варианту составят:

$$З_6 = 510 \cdot 806,4 \cdot \left(\frac{1}{0,74}\right) = 555\,762,2 \text{ руб./год}$$

Затраты при использовании разработанной программы рассчитываются по формуле:

$$З_9 = (T_r \cdot C_m + З_{\text{п}})/T_c, \quad (24)$$

где T_r – время, отводимое на работу с программой;

C_m – стоимость одного машинного часа;

$З_{\text{п}}$ – эксплуатационные затраты при использовании ПО;

T_c – срок службы программного обеспечения.

Цена машино-часа измеряется руб./час. и рассчитывается по формуле:

$$C_m = (Z_a + Z_m + Z_{тр})/T_{пер}, \quad (25)$$

где Z_a – затраты на амортизацию, руб.;

$Z_{тр}$ – затраты на текущий ремонт компьютера, руб.;

Z_m – затраты на материалы, руб.;

$T_{пер}$ – период разработки программного продукта.

Стоимость 1 машино-часа составит:

$$C_m = (1381,4 + 924 + 114,7)/(35 * 8) = 8,64 \text{ руб./час.}$$

Эксплуатационные расходы на разработку – общие расходы на разработку программного продукта. $Z_n = 92\,878,7$ руб.

Срок службы программы в среднем равен $T_c = 5$ лет.

Затраты при использовании программы будут равны:

$$Z_3 = (2016 \cdot 8,64 + 92\,878,7)/5 = 22\,060,6 \text{ руб./год}$$

Значение экономической эффективности от использования программного продукта будет составлять:

$$\mathcal{E} = 555\,762,2 - 22\,060,6 = 533\,701,6 \text{ руб./год}$$

После определения годового экономического эффекта необходимо рассчитать срок окупаемости затрат на разработку проекта по формуле:

$$T_{ок} = \frac{K}{\mathcal{E}}, \quad (26)$$

где K – затраты на разработку проекта, руб.;

Э – годовая эффективность, руб.

Срок окупаемости затрат на разработку программного продукта составит:

$$T_{\text{ок}} = \frac{92\,878,7}{533\,701,6} = 0,17 \text{ года} = 2,04 \text{ месяца}$$

Таким образом, срок окупаемости проекта составляет примерно 2 месяца. Малый срок объясняется тем, что проект не включает сложных программных разработок.

Коэффициент экономической эффективности разработки ($E\phi$):

$$E_{\phi} = \frac{\text{Э}}{\text{К}} \quad (27)$$

Нормативное значение коэффициента эффективности капитальных вложений $E_n = 0,33$, если $E\phi > E_n$, то делается вывод об эффективности капитальных вложений.

Коэффициент экономической эффективности разработки ($E\phi$) равен:

$$E_{\phi} = \frac{533\,701,6}{92\,878,7} = 5,7$$

Так как $E\phi = 5,7 > E_n$, то разработка и внедрение разработанного продукта является эффективной, т.е. эффект от использования данной программы окупает все затраты, связанные с проектированием и эксплуатацией. В таблице 15 приведены сводные данные экономического обоснования разработки и внедрения проекта.

Таблица 15 – Таблица показателей экономической эффективности

Показатель	Величина
Трудозатраты на разработку проекта, чел*час	263,3
Затраты на разработку проекта, руб.	92 878,7
Экономическая эффективность проекта, руб.	555 762,2
Коэффициент экономической эффективности	5,7
Срок окупаемости, мес.	2

Таким образом, произведенный экономический анализ эффективности создания и эксплуатации программ доказывает целесообразность его использования в Абаканском отделении №8602 Сбербанка России.

3.4 Выводы по экономическому разделу «Расчет показателей экономической эффективности от внедрения потребительского калькулятора в Абаканском отделении №8602 Сбербанка России»

Общие трудозатраты на разработку программ составляют 263,3 чел*час.

Затраты на разработку программ составляют 92 878,7 рублей.

Экономическая эффективность от использования увеличивается по сравнению с базовым вариантом и составляет 555 762,2 руб./год.

Срок окупаемости проекта составляет примерно 2 месяца. Малый срок объясняется тем, что проект не включает сложных программных разработок.

Так как $E\phi = 5,7 > E_n$, то разработка и внедрение разработанного продукта является эффективным, т.е. эффект от использования данной программы окупает все затраты, связанные с проектированием и эксплуатацией.

До внедрения разработанной программы «Потребительский калькулятор» время работы сотрудников данной организации составляло 209,95 часов в месяц, после внедрения временные затраты сократятся до 35 часов в месяц.

ЗАКЛЮЧЕНИЕ

Целью выпускной квалификационной работы является улучшение качества обслуживания клиентов Абаканского отделения №8602 Сбербанка России за счет разработки потребительского калькулятора.

Проанализирована деятельность организации ПАО «Сбербанк России». Построены диаграммы вариантов использования отделения кредитования Абаканского отделения №8602.

Проведен поиск и анализ существующих подобных программных продуктов: онлайн-калькулятор потребительского кредита для ПК и «Универсальный калькулятор» для платформ Android и iOS. В результате анализа выявлено, что существующие калькуляторы не удовлетворяют потребностям ПАО «Сбербанк России», и принято решение собственной разработки программного продукта в двух версиях: для специалистов (приложение для ПК) и потребителей (мобильное приложение).

В качестве среды разработки мобильного приложения выбран Android Studio. Основным языком программирования был выбран JavaScript. Для создания интерфейса и оформления внешнего вида приложения использовались такие языки, как HTML и CSS.

Для разработки стационарной версии калькулятора для специалистов использовалась интегрированная среда разработки Delphi 7.

Разработаны два программных продукта для ПК и платформы Android.

Дано экономическое обоснование эффекта от внедрения потребительского калькулятора в ПАО «Сбербанк России».

Разработанные калькуляторы для организации ПАО «Сбербанк России» позволят сэкономить время работы не только специалиста кредитного отдела, но и потребителя. Мобильная версия калькулятора для потребительского кредита предусмотрена для использования физическими лицами: создан удобный и простой интерфейс приложения, который имеет

основные необходимые функции для расчета кредита. Также для улучшения работы продуманы подсказки.

Цель и задачи, поставленные для решения проблемы организации ПАО «Сбербанк России» достигнуты.

СПИСОК СОКРАЩЕНИЙ

ПАО – публичное акционерное общество.

UML (англ. Unified Modeling Language) — унифицированный язык моделирования.

DFD (англ. Data Flow Diagrams) – диаграммы потоков данных.

ПК – персональный компьютер.

MS (полное название Microsoft Office) – программы, разработанные компанией Microsoft.

HTML (англ. HyperText Markup Language) — язык гипертекстовой разметки.

XML (англ. Extensible Markup Language) расширяемый язык разметки.

Js. (полное название JavaScript) — это легкий, интерпретируемый, объектно – ориентированный язык с функциями первого класса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бебрыш, Т. Н. Оценка экономической эффективности разработки программного продукта : метод.указания к выполнению экономического раздела дипломного проекта / сост. Т. Н. Бебрыш. – Абакан:Сиб. федер. ун-т; ХТИ – филиал СФУ, 2009. – 34 с.
2. Информационный ресурс, посвященный языкамHTML иCSS [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/>.
3. Информационный ресурс, посвященный языку JavaScript и смежным технологиям [Электронный ресурс]. – Режим доступа: <http://javascript.ru/>.
4. Информация об архитектуре приложений [Электронный ресурс]. – Режим доступа: <http://bourabai.ru/dbt/richclient.htm>.
5. Кабаева, Е. В. Оформление текстовых и графических материалов при подготовке курсовых и дипломных проектов: метод.указания / сост. Е. В. Кабаева. – Абакан: Сиб, федер, ун-т; ХТИ – филиал СФУ, 2009. – 36 с.
6. Кокова, В. И. Базы данных: учебное пособие. Для студентов очной и заочной форм обучения направления подготовки бакалавров 230700.62 – «Прикладная информатика» / В. И. Кокова. – Абакан: Сибирский федеральный университет; ХТИ – филиал СФУ, 2014. – 211 с.
7. Минеев, П. В. Выполнение и защита выпускной квалификационной работы по направлению подготовки 09.03.03 – Прикладная информатика : методические указания / П. В. Минеев; Сиб. федер. ун-т, ХТИ – филиал СФУ. – Абакан: Ред.-изд. сектор ХТИ – филиала СФУ, 2014. – 41 с.
8. Минеев, П. В. Дипломное проектирование: метод.указания / сост. П. В. Минеев. – Абакан:Сиб. федер ун-т; ХТИ – филиал СФУ, 2009. – 52 с.

9. Минеев, П. В. Проектирование информационных систем: учебно-методическое пособие для студентов специальности 080801.65 «Прикладная информатика(в экономике)» всех форм обучения / Сост. П. В. Минеев. – Красноярск: КГТУ, 2005. – 68 с.
10. Минеев, П. В. Структурное проектирование информационных систем: методические указания к лабораторным и расчетно-графическим работам / Сост. П. В. Минеев. – Красноярск : КГТУ, 2006. – 52 с.
11. Сайт фреймворка Bootstrap [Электронный ресурс]. – Режим доступа: <http://getbootstrap.com/>.
12. Сбербанк России [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Сбербанк_России.
13. Специализированный сайт Хабрахабр [Электронный ресурс]: содержит информацию в сфере IT-технологий. – Режим доступа: <http://habrahabr.ru/>.
14. Статьи по программированию [Электронный ресурс]. – Режим доступа: <http://on-line-teaching.com/>.
15. СТО 4.2–07–2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности, Красноярск: ИПК СФУ, 2014. – 60 с.
16. Формулы ежемесячных платежей [Электронный ресурс]. – Режим доступа: <http://exceltable.com/otchety/kalkulyator-rascheta-kredita>.
17. Форум о кредитах и займах [Электронный ресурс]. – Режим доступа: <http://crediteuro.ru/forum>.
18. Форум программистов Delphi7 [Электронный ресурс]. – Режим доступа: <http://www.cyberforum.ru/delphi/>.
19. Харитонов, И.А. MicrosoftAccess 2000 / И.А. Харитонов. – СПб.: БХВ – Санкт-Петербург, 1999. – 488 с.

ПРИЛОЖЕНИЕ А

Программный код файла-индекса index.html

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Кредитный калькулятор</title>

    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col-sm-offset-1 col-sm-10 col-md-offset-2 col-md-8 col-lg-
offset-3 col-lg-6">
                <div class="tab-content">
                    <div class="tab-pane fade in active" id="tab-welcome">
                        <div class="text-center">
                            <h4>Потребительский калькулятор, для расчета
кредита</h4>

                            <div class="display-inline-block margin-
top-15">
                                <p>Выберите вид платежа:</p>
                                <button type="button" class="btn
btn-default btn-block" data-toggle="tab" data-target="#tab-kredit" data-kredit-
type="annuitentnyi">Аннуитентный платеж</button>
                                <button type="button" class="btn
btn-default btn-block" data-toggle="tab" data-target="#tab-kredit" data-kredit-
type="differencialnyi">Дифференцированный платеж</button>
                                <br>
                                <p>Дополнительный расчет:</p>
                                <button type="button" class="btn
btn-default btn-block" data-toggle="tab" data-target="#tab-penya">Расчет суммы
пеней</button>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="tab-pane fade" id="tab-kredit">
        <div class="clearfix">
```

```

        <span class="pull-right back-link" data-
toggle="tab" data-target="#tab-welcome"><span class="glyphicon glyphicon-home margin-
right-5"></span> К экрану приветствия</span>
    </div>

    <h5 class="js-title">Заголовок устанавливается
автоматически</h5>

    <form>
        <div class="form-group">
            <label class="control-label"
for="summa-kredita">Сумма кредита (руб.):</label>
            <input type="number" data-
type="int" class="form-control input-sm" id="summa-kredita" value="">
        </div>

        <div class="form-group js-tab-controls">
            <label class="radio-inline">
                <input type="radio"
name="kredit-controls" class="js-show-srok-kredita" checked> Срок кредита
            </label>
            <label class="radio-inline">
                <input type="radio"
name="kredit-controls" class="js-show-ezhem-platezh"> Ежемес. платеж
            </label>
        </div>

        <div class="js-group-srok-kredita form-
group">
            <label class="control-label"
for="srok-kredita">Срок кредита (мес.):</label>
            <input type="number" data-
type="int" class="form-control input-sm" id="srok-kredita" value="">
        </div>

        <div class="js-group-ezhem-platezh form-
group hide">
            <label class="control-label"
for="ezhem-platezh">Ежемесячный платеж (руб.):</label>
            <input type="number" data-
type="float" class="form-control input-sm" id="ezhem-platezh" value="">
        </div>

        <div class="form-group">
            <label class="control-label"
for="procentnaya-stavka">Процентная ставка (% в год):</label>
            <input type="number" data-
type="float" class="form-control input-sm" id="procentnaya-stavka" value="">
        </div>

        <div class="form-group">

```

```

for="start-date">Дата получения кредита:</label>
class="form-control input-sm" id="start-date" value="">
</div>

value="">

<div class="form-group text-center">
  <button type="button" class="btn
btn-primary btn-sm js-calc-kredit">Рассчитать</button>
  <button type="button" class="btn
btn-default btn-sm js-reset-kredit">Сбросить</button>

  <button type="button" class="btn
btn-default btn-sm js-ann-diff" data-toggle="popover" title="Аннуитентный платеж" data-
content="При аннуитентном порядке погашения, ежемесячные выплаты равны в течение всего срока
кредитования." data-placement="top" data-trigger="focus" data-template='<div class="popover
margin-right-15"><div class="arrow"></div><h3 class="popover-title"></h3><div
class="popover-content"></div></div>'><span class="glyphicon glyphicon-question-
sign"></span></button>

  <button type="button" class="btn
btn-default btn-sm js-help-diff" data-toggle="popover" title="Дифференцированный платеж" data-
content="При дифференцированном порядке погашения, ежемесячные выплаты уменьшаются в течение
всего срока кредитования." data-placement="top" data-trigger="focus" data-template='<div
class="popover margin-right-15"><div class="arrow"></div><h3 class="popover-
title"></h3><div class="popover-content"></div></div>'><span class="glyphicon glyphicon-
question-sign"></span></button>

</div>
</form>

<div class="js-kredit-result"></div>
</div>

<div class="tab-pane fade" id="tab-penya">
  <div class="clearfix">
    <span class="pull-right back-link" data-
toggle="tab" data-target="#tab-welcome"><span class="glyphicon glyphicon-home margin-
right-5"></span> К экрану приветствия</span>
  </div>

  <h5 class="js-title">Заголовок устанавливается
автоматически</h5>

  <form>
    <div class="form-group">
      <label class="control-label"
for="summa-zadolzhen">Сумма просроченной задолженности (руб.):</label>

```

```

                                <input type="number" data-
type="float" class="form-control input-sm" id="summa-zadolzhen" value="">
                                </div>

                                <div class="form-group">
                                    <label class="control-label"
for="data-platezha">Дата платежа по договору:</label>
                                    <input type="date" data-type="date"
class="form-control input-sm" id="data-platezha" value="">
                                    </div>

                                <div class="form-group">
                                    <label class="control-label"
for="data-podscheta">Дата погашения:</label>
                                    <input type="date" data-type="date"
class="form-control input-sm" id="data-podscheta" value="">
                                    </div>

                                <div class="form-group">
                                    <label class="control-label"
for="penya">Пеня в день (%):</label>
                                    <input type="number" data-
type="float" class="form-control input-sm" id="penya" value="">
                                    </div>

                                <input type="hidden" id="kredit-type"
value="">

                                <div class="form-group text-center">
                                    <button type="button" class="btn
btn-primary btn-sm js-calc-penya">Рассчитать</button>
                                    <button type="button" class="btn
btn-default btn-sm js-reset-penya">Сбросить</button>
                                    </div>
                                </form>

                                <div class="js-penya-result"></div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>

            <script src="js/jquery.min.js"></script>
            <script src="js/bootstrap.min.js"></script>
            <script src="js/script.js"></script>
        </body>
    </html>

```


ПРИЛОЖЕНИЕ Б

Программный код файла script.js

```
function inputHasErrors(input) {
    var result = false;

    var $parent_form_group = $(input).closest('.form-group');
    $parent_form_group.removeClass('has-error');

    if ($(input).val() === '') {
        $parent_form_group.addClass('has-error');
        result = true;
    }

    if ($(input).data('type') === 'int') {
        var value = parseFloat($(input).val());

        if (Math.round(value) !== value) {
            $parent_form_group.addClass('has-error');
            result = true;
        }
    }

    if ($(input).data('type') === 'float') {
        var value = parseFloat($(input).val());

        if (!Number.isFinite(value)) {
            $parent_form_group.addClass('has-error');
            result = true;
        }
    }

    if ($(input).data('type') === 'date') {
        if (!Date.parse($(input).val())) {
            $parent_form_group.addClass('has-error');
            result = true;
        }
    }

    return result;
}

function getFormattedDate(date) {
    var now = date || new Date();

    return now.getFullYear() + '-' +
        ('0' + (now.getMonth() + 1)).slice(-2) + '-' +
        ('0' + now.getDate()).slice(-2);
}

function getUserFormattedDate(date) {
    var now = date || new Date();

    return ('0' + now.getDate()).slice(-2) + '.' +
        ('0' + (now.getMonth() + 1)).slice(-2) + '.' +
        ('0' + now.getDate()).slice(-2);
}
```

```

        now.getFullYear();
    }

    function setKreditDefaults() {
        $('#summa-kredita').val(""); // 100000
        $('#srok-kredita').val(""); // 6
        $('#ezhem-platezh').val(""); // 17156.14
        $('#procentnaya-stavka').val(""); // 10
        $('#start-date').val(getFormattedDate());

        $('#js-kredit-result').html("");
    }

    function setPenyaDefaults() {
        var data_platezha = new Date();
        data_platezha.setDate(data_platezha.getDate() - 29); // 29

        $('#summa-zadolzhen').val(""); // 1200
        $('#data-platezha').val(getFormattedDate(data_platezha));
        $('#data-podscheta').val(getFormattedDate());
        $('#penya').val(""); // 20

        $('#js-penya-result').html("");
    }

    function kreditHasErrors(){
        result_1 = inputHasErrors($('#summa-kredita'));

        if (!$('.js-group-srok-kredita').hasClass('hide')) {
            result_2 = inputHasErrors($('#srok-kredita'));
        } else if (!$('.js-group-ezhem-platezh').hasClass('hide')) {
            result_2 = inputHasErrors($('#ezhem-platezh'));
        }

        result_3 = inputHasErrors($('#procentnaya-stavka'));
        result_4 = inputHasErrors($('#start-date'));

        return (result_1 || result_2 || result_3 || result_4);
    }

    function penyaHasErrors(){
        result_1 = inputHasErrors($('#summa-zadolzhen'));
        result_2 = inputHasErrors($('#data-platezha'));
        result_3 = inputHasErrors($('#data-podscheta'));
        result_4 = inputHasErrors($('#penya'));

        if (result_5 = !(Date.parse($('#data-podscheta').val()) > Date.parse($('#data-platezha').val()))) {
            $('#data-platezha').closest('.form-group').addClass('has-error');
        }

        return result_1 || result_2 || result_3 || result_4 || result_5;
    }

    function calculateKredit() {
        var variant;
        if (!$('.js-group-srok-kredita').hasClass('hide')) {
            variant = 'srok-kredita';
        } else if (!$('.js-group-ezhem-platezh').hasClass('hide')) {

```

```

        variant = 'ezhem-platezh';
    }

    var summa_kredita = parseInt($('#summa-kredita').val(), 10);
    var procentnaya_stavka = parseFloat($('#procentnaya-stavka').val());
    var start_date = new Date(Date.parse($('#start-date').val()));
    var kredit_type = $('#kredit-type').val();

    var osnovn_platezh, ezhem_platezh, srok_kredita;

    var month_rate = procentnaya_stavka / 100 / 12; // месячная процентная ставка по
кредиту (= годовая ставка / 12)

    if ((kredit_type === 'differencialnyi') || (kredit_type === 'annuitentnyi' && variant ===
'srok-kredita')) {
        srok_kredita = parseInt($('#srok-kredita').val(), 10);
    } else if (kredit_type === 'annuitentnyi' && variant === 'ezhem-platezh') {
        ezhem_platezh = parseFloat($('#ezhem-platezh').val());

        srok_kredita = Math.round(Math.log(-ezhem_platezh / (summa_kredita *
month_rate + ezhem_platezh)) / Math.log(1 + month_rate));
    }

    var end_date = new Date(start_date.getTime());
    end_date.setMonth(end_date.getMonth() + srok_kredita);

    if (kredit_type === 'annuitentnyi') {

        ezhem_platezh = summa_kredita * month_rate * (Math.pow(1 + month_rate,
srok_kredita)) / (Math.pow(1 + month_rate, srok_kredita) - 1);
        var summa_platezey = ezhem_platezh * srok_kredita;

    } else if (kredit_type === 'differencialnyi') {

        osnovn_platezh = summa_kredita / srok_kredita;

        var summa_platezey = 0;
        for (var i = 0; i < srok_kredita; i++) {
            summa_platezey += osnovn_platezh + (summa_kredita -
(osnovn_platezh * i)) * month_rate;
        }

    }

    var pereplata = summa_platezey - summa_kredita;

    var output =
        '<div class="well well-sm well-result clearfix">' +
        '<div>Сумма кредита: <span class="pull-right">' + summa_kredita + '
руб.</span></div>' +
        '<div>Срок кредита: <span class="pull-right">' + srok_kredita + '
мес.</span></div>' +
        '<div>Процентная ставка: <span class="pull-right">' + procentnaya_stavka
+ '%</span></div>' +
        '<div>Дата получения кредита: <span class="pull-right">' +
getUserFormattedDate(start_date) + '</span></div>' +
        '<div>Дата последней выплаты: <span class="pull-right">' +
getUserFormattedDate(end_date) + '</span></div>';

```

```

        if (kredit_type === 'annuitentnyi') {
            output += '<div>Ежемесячный платеж: <span class="pull-right"><strong>' +
                ezhem_platezh.toFixed(2) + ' руб.</strong></span></div>';
        }

        output +=
            '<div>Полная сумма выплат: <span class="pull-right"><strong>' +
            summa_platezey.toFixed(2) + ' руб.</strong></span></div>' +
            '<div>Переплата: <span class="pull-right"><strong>' +
            pereplata.toFixed(2) + ' руб.</strong></span></div>' +
            '</div>';

        output +=
            '<div class="table-responsive">' +
            '<table class="table table-bordered table-condensed text-center">' +
            '<thead>' +
            '<tr>' +
            '<th class="text-center">№</th>' +
            '<th class="text-center">Дата<br>платежа</th>' +
            '<th class="text-center">Остаток<br>кредита</th>' +
            '<th class="text-center">Проценты</th>' +
            '<th class="text-center">Основной<br>платеж</th>' +
            '<th class="text-center">Всего<br>за платеж</th>' +
            '</tr>' +
            '</thead>' +
            '<tbody>';

        var date = new Date(start_date.getTime());
        var ostatok_po_kreditu = summa_kredita;
        var percent_pay, credit_pay;

        for (var i = 0; i < srok_kredita; i++) {
            date.setMonth(date.getMonth() + 1);

            if (kredit_type === 'annuitentnyi') {
                percent_pay = ostatok_po_kreditu * month_rate;
                osnovn_platezh = ezhem_platezh - percent_pay;
                credit_pay = osnovn_platezh;
            } else if (kredit_type === 'differencialnyi') {
                percent_pay = ostatok_po_kreditu * month_rate;
                ezhem_platezh = osnovn_platezh + percent_pay;
                credit_pay = osnovn_platezh;
            }

            output += '<tr>';
            output += '<td>' + (i + 1) + '</td>';
            output += '<td>' + getUserFormattedDate(date) + '</td>';
            output += '<td>' + ostatok_po_kreditu.toFixed(2) + '</td>';
            output += '<td>' + percent_pay.toFixed(2) + '</td>';
            output += '<td>' + credit_pay.toFixed(2) + '</td>';
            output += '<td>' + ezhem_platezh.toFixed(2) + '</td>';
            output += '</tr>';

            ostatok_po_kreditu -= credit_pay;
        }

        output += '</tbody></table></div>';

        $('<js-kredit-result>').html(output);

```

```

}

function calculatePenya() {
    var summa_zadolzhen = parseFloat($('#summa-zadolzhen').val());
    var data_platezha = new Date(Date.parse($('#data-platezha').val()));
    var data_podscheta = new Date(Date.parse($('#data-podscheta').val()));
    var penya = parseFloat($('#penya').val());

    var kolich_dnei = Math.abs(data_podscheta.getTime() - data_platezha.getTime());
    kolich_dnei = Math.ceil(kolich_dnei / (1000 * 60 * 60 * 24));

    var summa_penei = summa_zadolzhen * penya / 100 * kolich_dnei;

    $('#js-penya-result').html(
        '<div class="well well-sm well-result">' +
        '<div>Сумма задолженности: <span class="pull-right">' +
summa_zadolzhen + ' руб.</span></div>' +
        '<div>Дата платежа по договору: <span class="pull-right">' +
getUserFormattedDate(data_platezha) + '</span></div>' +
        '<div>Дата погашения: <span class="pull-right">' +
getUserFormattedDate(data_podscheta) + '</span></div>' +
        '<div>Количество дней просрочки: <span class="pull-right">' + kolich_dnei
+ '</span></div>' +
        '<div>Размер пени в день: <span class="pull-right">' + penya +
'%</span></div>' +
        '<div>Сумма пеней: <span class="pull-right"><strong>' +
summa_penei.toFixed(2) + ' руб.</strong></span></div>' +
        '</div>'
    );
}

/* Когда документ загрузился, выполняется следующий код */
$(document).on('ready', function (){

    $('[data-toggle="popover"]').popover();

    /* При клике на кнопку на экране приветствия устанавливается заголовок открываемой вкладки
    */
    $('#tab-welcome button').on('click', function(){
        $('#tab-welcome button').text($('#tab-welcome button').text());
        $('#kredit-type').val($('#tab-welcome button').text());

        if ($('#kredit-type') === 'differencialnyi') {
            $('#js-tab-controls').hide();
            $('#js-ann-diff').hide();
            $('#js-help-diff').show();
        } else {
            $('#js-tab-controls').show();
            $('#js-ann-diff').show();
            $('#js-help-diff').hide();
        }

        $('#js-show-srok-kredita').click();

        setKreditDefaults();
        setPenyaDefaults();
    });

    /* Когда поле ввода теряет фокус, то вызывается проверка введенных в него данных */

```

```

$('.form-control').on('blur', function(){
    inputHasErrors(this);
});

/* Вычисление кредита */
$('.js-calc-kredit').on('click', function(){
    $('.js-kredit-result').html("");
    if (!kreditHasErrors()) {
        calculateKredit();
    }
});

/* Сброс формы кредита */
$('.js-reset-kredit').on('click', function(){
    $(this).closest('form').find('.form-group').removeClass('has-error');
    setKreditDefaults();
});

/* Вычисление пени */
$('.js-calc-penya').on('click', function(){
    $('.js-penya-result').html("");
    if (!penyaHasErrors()) {
        calculatePenya();
    }
});

/* Сброс формы пени */
$('.js-reset-penya').on('click', function(){
    $(this).closest('form').find('.form-group').removeClass('has-error');
    setPenyaDefaults();
});

$('.js-show-srok-kredita').on('click', function(){
    if ($('.js-group-srok-kredita').hasClass('hide')) {
        $('.js-group-srok-kredita').removeClass('hide');
        $('.js-group-ezhem-platezh').addClass('hide');
    }
});

$('.js-show-ezhem-platezh').on('click', function(){
    if ($('.js-group-ezhem-platezh').hasClass('hide')) {
        $('.js-group-ezhem-platezh').removeClass('hide');
        $('.js-group-srok-kredita').addClass('hide');
    }
});
});

```

ПРИЛОЖЕНИЕ В

Файл стилей style.

```
* {
    outline: none !important
}

body {
    background-color: #ffff0;
    padding-top: 15px;
    padding-bottom: 15px;
}

label {
    margin-bottom: 0;
}

th {
    vertical-align: middle !important;
}

.back-link {
    border-bottom: 1px dashed #666;
}

.well-result {
    background-color: #ffffcc;
    border-radius: 4px;
}

@media (max-width: 479px) {
    #tab-welcome {
        padding-top: 50px;
    }
}

.margin-top-15 {margin-top: 15px;}
.margin-right-5 {margin-right: 5px;}
.margin-right-15 {margin-right: 15px;}
.display-inline-block {display: inline-block;}
```

ПРИЛОЖЕНИЕ Г

Листинг модуля приложения «Потребительский калькулятор» (файл Main.pas)

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Buttons, XPMAN, ComCtrls;

type
  TfrmMain = class(TForm)
    pnlHeader: TPanel;
    pnlControls: TPanel;
    pnlTabCredit: TPanel;
    sbSwitchTabAnn: TSpeedButton;
    sbSwitchTabDiff: TSpeedButton;
    Label1: TLabel;
    Label2: TLabel;
    sbSwitchTabPenya: TSpeedButton;
    XPMANifest: TXPMANifest;
    pnlTabPenya: TPanel;
    btnPenyaCalc: TButton;
    pnlTabWelcome: TPanel;
    Label3: TLabel;
    Image1: TImage;
    Label4: TLabel;
    Label5: TLabel;
    lblPenyaHeader: TLabel;
    edtPenyaSumma: TEdit;
    lblPenaySumma: TLabel;
    edtPenyaPercent: TEdit;
    lblPenyaDataPlatezha: TLabel;
    dtpPenyaDataPlateza: TDateTimePicker;
    Label6: TLabel;
    dtpPenyaDataPodscheta: TDateTimePicker;
    Label7: TLabel;
    btnPenyaClear: TButton;
    btnCreditCalc: TButton;
    lblCreditHeader: TLabel;
    edtCreditSumma: TEdit;
    Label9: TLabel;
    edtCreditPercent: TEdit;
    lblCreditSrok: TLabel;
    Label11: TLabel;
    dtpCreditStartDate: TDateTimePicker;
    Label12: TLabel;
    btnCreditClear: TButton;
    edtCreditSrok: TEdit;
    edtCreditType: TEdit;
    rbShowCreditSrok: TRadioButton;
    rbShowEzhemPlatezh: TRadioButton;
    lblCreditEzhemPlatezh: TLabel;
    edtCreditEzhemPlatezh: TEdit;
    procedure sbSwitchTabAnnClick(Sender: TObject);
    procedure sbSwitchTabDiffClick(Sender: TObject);
    procedure sbSwitchTabPenyaClick(Sender: TObject);
    procedure btnPenyaClearClick(Sender: TObject);
    procedure btnCreditClearClick(Sender: TObject);
    procedure btnPenyaCalcClick(Sender: TObject);
    procedure btnCreditCalcClick(Sender: TObject);
```



```

    procedure rbShowCreditsrokClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmMain: TfrmMain;

implementation

uses DateUtils, FormPenya, FormCredit, Math, StrUtils;

{$R *.dfm}

procedure setCreditDefaults();
begin
    with frmMain do
    begin
        edtCreditSumma.Text := '';           // 100000
        edtCreditSrok.Text := '';           // 6
        edtCreditEzhemPlatezh.Text := '';   // 17156,14
        edtCreditPercent.Text := '';        // 10
        dtpCreditStartDate.Date := Now;
    end;
end;

procedure setPenyaDefaults();
begin
    with frmMain do
    begin
        edtPenyaSumma.Text := '';           // 1200
        edtPenyaPercent.Text := '';         // 20
        dtpPenyaDataPodscheta.Date := Now;
        dtpPenyaDataPlateza.Date := IncMonth(Now, -1);
    end;
end;

procedure calculatePenya();
var
    summa, penya, summa_penei: Currency;
    data_platezha, data_podscheta: TDateTime;
    kolich_dney: Integer;
begin
    with frmMain do
    begin
        summa := StrToCurr(edtPenyaSumma.Text);
        data_platezha := dtpPenyaDataPlateza.Date;
        data_podscheta := dtpPenyaDataPodscheta.Date;
        penya := StrToCurr(edtPenyaPercent.Text);
    end;

    kolich_dney := DaysBetween(data_platezha, data_podscheta);

    summa_penei := summa * penya / 100 * kolich_dney;

    with frmPenyaResult do
    begin
        lblSumma.Caption := CurrToStrF(summa, ffNumber, 2) + ' py6.';
        lblDataPlatezha.Caption := DateToStr(data_platezha);
        lblDataPodscheta.Caption := DateToStr(data_podscheta);
        lblKolichDney.Caption := IntToStr(kolich_dney);
        lblPenyaPercent.Caption := CurrToStrF(penya, ffNumber, 2) + '%';
        lblSummaPenei.Caption := CurrToStrF(summa_penei, ffNumber, 2) + ' py6.';
    end;
end;

procedure calculateCredit();
var

```

```

summa, procentnaya_stavka, month_rate, osnovn_platezh, ezhem_platezh,
    summa_platezey, pereplata,
    ostatok_po_kreditu, percent_pay, credit_pay: Real;
start_date, end_date: TDateTime;
srok_kredita, I: Integer;
kredit_type, variant: String;
begin
    with frmMain do
        begin
            variant := IfThen(edtCreditsrok.Visible, 'srok-kredita', 'ezhem-
platezh');

            summa := StrToCurr(edtCreditSumma.Text);
            procentnaya_stavka := StrToCurr(edtCreditPercent.Text);
            start_date := dtpCreditStartDate.Date;
            kredit_type := edtCreditType.Text;
        end;

        month_rate := procentnaya_stavka / 100 / 12;

        if (kredit_type = 'differencirovannyi')
            OR ((kredit_type = 'annuitentnyi') AND (variant = 'srok-kredita')) then
            begin
                srok_kredita := StrToInt(frmMain.edtCreditsrok.Text);
            end
        else if (kredit_type = 'annuitentnyi') AND (variant = 'ezhem-platezh')
        then
            begin
                ezhem_platezh := StrToCurr(frmMain.edtCreditEzhemPlatezh.Text);
                srok_kredita := Round(
                    Ln(-ezhem_platezh / (summa * month_rate + -ezhem_platezh))
                    / Ln(1 + month_rate));
            end;

            end_date := IncMonth(start_date, srok_kredita);

            if (kredit_type = 'annuitentnyi') then
            begin
                ezhem_platezh := summa * month_rate *
                    Power(1 + month_rate, srok_kredita) / (Power(1 + month_rate,
srok_kredita) - 1);
                summa_platezey := ezhem_platezh * srok_kredita;
            end
            else if (kredit_type = 'differencirovannyi') then
            begin
                osnovn_platezh := summa / srok_kredita;

                summa_platezey := 0;
                for I := 1 to srok_kredita do
                    begin
                        summa_platezey := summa_platezey +
                            (osnovn_platezh + (summa - (osnovn_platezh * I)) * month_rate);
                    end;
                end;

            end;

            pereplata := summa_platezey - summa;

            with frmCreditResult do
                begin
                    Caption := frmMain.lblCreditHeader.Caption;
                    lblCreditHeader.Caption := frmMain.lblCreditHeader.Caption;
                    lblSumma.Caption := CurrToStrF(summa, ffNumber, 2) + ' py6.';
                    lblCreditsrok.Caption := IntToStr(srok_kredita) + ' mec.';
                    lblProcentnayaStavka.Caption := FloatToStr(procentnaya_stavka) + '%';
                    lblStartDate.Caption := DateToStr(start_date);
                    lblEndDate.Caption := DateToStr(end_date);
                    lblSummaPlatezhey.Caption := CurrToStrF(summa_platezey, ffNumber, 2) + '
py6.';
                    lblPereplata.Caption := CurrToStrF(pereplata, ffNumber, 2) + ' py6.';
                end;
            end;
        end;
    end;
end;

```

```

    if (kredit_type = 'annuitentnyi') then
    begin
        lblEzhemPlatezhDescr.Visible := true;
        lblEzhemPlatezh.Visible := true;
        lblEzhemPlatezh.Caption := CurrToStrF(ezhem_platezh, ffNumber, 2) + '
py6.';
    end
    else if (kredit_type = 'differencirovannyi') then
    begin
        lblEzhemPlatezhDescr.Visible := false;
        lblEzhemPlatezh.Visible := false;
    end;

    StringGrid1.RowCount := 1;
    StringGrid1.Colwidths[0] := 30;

    StringGrid1.Cells[0, 0] := '№';
    StringGrid1.Cells[1, 0] := 'Дата платежа';
    StringGrid1.Cells[2, 0] := 'Остаток кредита';
    StringGrid1.Cells[3, 0] := 'Проценты';
    StringGrid1.Cells[4, 0] := 'Основной платеж';
    StringGrid1.Cells[5, 0] := 'Всего за платеж';

    ostatok_po_kreditu := summa;

    for I := 1 to srok_kredita do
    begin
        if (kredit_type = 'annuitentnyi') then
        begin
            percent_pay := ostatok_po_kreditu * month_rate;
            osnovn_platezh := ezhem_platezh - percent_pay;
            credit_pay := osnovn_platezh;
        end
        else if (kredit_type = 'differencirovannyi') then
        begin
            percent_pay := ostatok_po_kreditu * month_rate;
            ezhem_platezh := osnovn_platezh + percent_pay;
            credit_pay := osnovn_platezh;
        end;

        StringGrid1.RowCount := StringGrid1.RowCount + 1;
        StringGrid1.Cells[0, I] := IntToStr(I);
        StringGrid1.Cells[1, I] := DateToStr(IncMonth(start_date, I));
        StringGrid1.Cells[2, I] := CurrToStrF(ostatok_po_kreditu, ffNumber,
2);
        StringGrid1.Cells[3, I] := CurrToStrF(percent_pay, ffNumber, 2);
        StringGrid1.Cells[4, I] := CurrToStrF(credit_pay, ffNumber, 2);
        StringGrid1.Cells[5, I] := CurrToStrF(ezhem_platezh, ffNumber, 2);

        ostatok_po_kreditu := ostatok_po_kreditu - credit_pay;
    end;
end;

end;

procedure TfrmMain.sbSwitchTabAnnClick(Sender: TObject);
begin
    setCreditDefaults();
    edtCreditType.Text := 'annuitentnyi';
    lblCreditHeader.Caption := (Sender as TSpeedButton).Caption;

    rbShowCreditSrok.Enabled := true;
    rbShowEzhemPlatezh.Enabled := true;

    pnlTabCredit.Visible := true;
    pnlTabPenya.Visible := false;
    pnlTabWelcome.Visible := false;
end;

procedure TfrmMain.sbSwitchTabDiffClick(Sender: TObject);

```

```

begin
    setCreditDefaults();
    edtCreditType.Text := 'differencirovannyi';
    lblCreditHeader.Caption := (Sender as TSpeedButton).Caption;

    rbShowCreditSrok.Checked := true;

    rbShowCreditSrok.Enabled := false;
    rbShowEzhemPlatezh.Enabled := false;

    pnlTabCredit.Visible := true;
    pnlTabPenya.Visible := false;
    pnlTabWelcome.Visible := false;
end;

procedure TfrmMain.sbSwitchTabPenyaClick(Sender: TObject);
begin
    pnlTabCredit.Visible := false;
    pnlTabPenya.Visible := true;
    pnlTabWelcome.Visible := false;

    setPenyaDefaults();
end;

procedure TfrmMain.btnCreditClearClick(Sender: TObject);
begin
    setCreditDefaults();
end;

procedure TfrmMain.btnPenyaClearClick(Sender: TObject);
begin
    setPenyaDefaults();
end;

procedure TfrmMain.btnPenyaCalcClick(Sender: TObject);
begin
    calculatePenya();
    frmPenyaResult.ShowModal;
end;

procedure TfrmMain.btnCreditCalcClick(Sender: TObject);
begin
    calculateCredit();
    frmCreditResult.ShowModal;
end;

procedure TfrmMain.rbShowCreditSrokClick(Sender: TObject);
begin
    lblCreditSrok.Visible := not lblCreditSrok.Visible;
    edtCreditSrok.Visible := not edtCreditSrok.Visible;
    lblCreditEzhemPlatezh.Visible := not lblCreditEzhemPlatezh.Visible;
    edtCreditEzhemPlatezh.Visible := not edtCreditEzhemPlatezh.Visible;
end;

end.

```

ПРИЛОЖЕНИЕ Д

Листинг модуля приложения «Потребительский калькулятор» (файл FormPenya.pas)

```
unit FormPenya;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TfrmPenyaResult = class(TForm)
    Bevel1: TBevel;
    btnClose: TButton;
    lblPenyaHeader: TLabel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    lblSumma: TLabel;
    lblDataPlatezha: TLabel;
    lblDataPodscheta: TLabel;
    lblKolichestvo: TLabel;
    lblPenyaPercent: TLabel;
    lblSummaPenei: TLabel;
    btnPrint: TButton;
    procedure btnCloseClick(Sender: TObject);
    procedure btnPrintClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmPenyaResult: TfrmPenyaResult;

implementation

{$R *.dfm}

procedure TfrmPenyaResult.btnCloseClick(Sender: TObject);
begin
  frmPenyaResult.Close;
end;

procedure TfrmPenyaResult.btnPrintClick(Sender: TObject);
begin
  frmPenyaResult.Print;
end;

end.
```

ПРИЛОЖЕНИЕ Е

Листинг модуля приложения «Потребительский калькулятор» (файл FormCredit.pas)

```
unit FormCredit;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Grids;

type
  TfrmCreditResult = class(TForm)
    Bevel1: TBevel;
    btnClose: TButton;
    lblCreditHeader: TLabel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    lblSumma: TLabel;
    lblCreditSrok: TLabel;
    lblProcentnayaStavka: TLabel;
    lblStartDate: TLabel;
    lblEndDate: TLabel;
    lblSummaPlatezhey: TLabel;
    Label7: TLabel;
    lblPereplata: TLabel;
    lblEzhemPlatezhDescr: TLabel;
    lblEzhemPlatezh: TLabel;
    StringGrid1: TStringGrid;
    btnPrint: TButton;
    procedure btnCloseClick(Sender: TObject);
    procedure btnPrintClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmCreditResult: TfrmCreditResult;

implementation

{$R *.dfm}

procedure TfrmCreditResult.btnCloseClick(Sender: TObject);
begin
  frmCreditResult.Close;
end;

procedure TfrmCreditResult.btnPrintClick(Sender: TObject);
begin
  frmCreditResult.Print;
end;

end.
```

Бакалаврская работа выполнена мной самостоятельно.
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в 1 экземпляре.

Библиография 19 наименований.

« _____ » июня 2016 г.

(дата)

(подпись)(ФИО)

А.А. Крылова